悟空返利接口文档

申瑞鸿鑫网络科技有限公司

2017-1-13

文档信息

|  |  |
| --- | --- |
| 文档名称 | 悟空返利接口文档 |
| 文件编号 |  |
| 编制人 |  |
| 保密级别 |  |

修改过程

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 负责人 | 概述 |
| V1.0 | 2017-01-13 | 黄杰 | 新建文档 |
|  | 2017-01-13 | 黄杰 | 新增淘宝API接口接入 |
|  | 2017-01-14 | 黄杰 | 新增本地服务端接口接入 |
|  | 2017-01-16 | 黄杰 | 修正错误内容，新增部分接口 |
|  |  |  |  |

评审过程

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 日期 | 评审者 | 概述 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

分发范围

# Android端接口

## 淘宝API

### 淘宝API公共请求参数和请求地址

**请求地址：**

|  |  |  |
| --- | --- | --- |
| **环境** | **HTTP请求地址** | **HTTPS请求地址** |
| 正式环境 | http://gw.api.taobao.com/router/rest | https://eco.taobao.com/router/rest |
| 沙箱环境 | http://gw.api.tbsandbox.com/router/rest | https://gw.api.tbsandbox.com/router/rest |

**公共参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| method | API接口名称 | String | 是 |
| app\_key | TOP分配给应用的AppKey | String | 是 |
| sign\_method | 签名的摘要算法，可选值为：hmac，md5 | String | 是 |
| sign | API输入参数签名结果 | String | 是 |
| timestamp | 时间戳 | String | 是 |
| format | 响应格式。默认为xml格式，可选值：xml，json | String | 否 |
| v | API协议版本，可选值：2.0 | String | 是 |

**说明：**请求地址和请求参数是调用淘宝所有API都必须传入的，并且每个API传入的这些参数基本都相同，所以实际调用时可以把请求地址和公共参数进行封装，调用不同API只需传入对应API所另需的参数即可。

**另：**上面两种环境请求地址中，使用淘宝沙箱环境需注册沙箱环境账号，但是一直都注册失败，所以Android客户端开发中均采用**正式环境**的**HTTP请求地址**；

公共参数仅列出了Android客户端开发中所用的必须参数，省略了淘宝API提供的可选参数，其中**sign\_method**采用**MD5**算法，**format**采用**json**格式。

### 获取淘宝后台商品的所有类目

|  |  |
| --- | --- |
| **承载协议：** | HTTP GET |
| **接口方向：** | Android客户端 –> 淘宝API后台 |
| **API:** | taobao.itemcats.get |
| **接口功能:** | 客户端通过调用淘宝API，获取淘宝后台供卖家发布商品的标准商品类目。 |
| **官方文档：** | http://open.taobao.com/docs/api.htm?spm=a219a.7629065.0.0.PA0AOR&apiId=122 |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| cids | 商品所属类目ID列表，用半角逗号(,)分隔 例如:(18957,19562,) (cids、parent\_cid至少传一个) | Number [] | 特殊可选 |
| fields | 需要返回的字段列表，默认返回：cid,parent\_cid,name,is\_parent。 | String [] | 可选 |
| parent\_cid | 父商品类目 id，0表示根节点, 传输该参数返回所有子类目。 (cids、parent\_cid至少传一个) | Number | 特殊可选 |

**请求方式：**

开发者须拼接请求地址和公共请求参数以及请求参数，在代码中发起HTTP GET请求即可。HTTP POST请求也可。

**响应参数：**

该API返回一个json格式数据，开发者可对其进行解析，选取需要的信息。详细信息见官方文档。

**说明：**此API暂时没有在代码中动态调用，仅手动获取过淘宝商品类目的所有子目录，并挑选热门的商品类目进行展示。

### 获取淘宝客类目商品

|  |  |
| --- | --- |
| **承载协议：** | HTTP GET |
| **接口方向：** | Android客户端 –> 淘宝API后台 |
| **API:** | [taobao.tbk.item.get](http://open.taobao.com/docs/api.htm?apiId=24515) |
| **接口功能:** | 客户端通过调用淘宝API，获取淘宝客推广商品以展示在APP中引导用户进行消费。 |
| **官方文档：** | http://open.taobao.com/docs/api.htm?spm=a219a.7629065.0.0.q627y6&apiId=24515 |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| fields | 需要返回的字段列表 | String [] | 必须 |
| q | 查询词，关键字 | String | 特殊可选 |
| cat | 后台类目ID，用”,”分割，最大10个，该ID可以通过taobao.itemcats.get接口获取到 | String | 特殊可选 |
| itemloc | 所在地 | String | 可选 |
| sort | 排序\_des（降序），排序\_asc（升序），销量（total\_sales），淘客佣金比率（tk\_rate）， 累计推广量（tk\_total\_sales），总支出佣金（tk\_total\_commi） | String | 可选 |
| page\_no | 第几页，默认：１ | number | 可选 |
| page\_size | 页大小，默认20，1~100 | number | 可选 |

**请求方式：**

开发者须拼接请求地址和公共请求参数以及请求参数，在代码中发起HTTP GET请求即可。HTTP POST请求也可。

**响应参数：**

该API返回一个json格式数据，开发者可对其进行解析，选取需要的信息。详细信息见官方文档。

**说明：**此API用于获取淘宝的分类商品，筛选淘宝客佣金比例或销量比较高的商品展示在APP分类栏目下。

### 关键词搜索淘宝商品

**说明：**该接口与[1.1.3.](#_获取淘宝客类目商品)使用的是同一个淘宝API，只需将参数“q”传入需要搜索的关键字即可。

## 百川电商SDK

APP需集成百川电商SDK。

### 淘宝授权登录

SDK已集成登录授权功能，在APP内需要登录淘宝账号的地方调用即可：

登录淘宝账号：

AlibcLogin alibcLogin = AlibcLogin.getInstance();

alibcLogin.showLogin(AliSdkMenuActivity.this, new AlibcLoginCallback() {

@Override

public void onSuccess() {

Toast.makeText(AliSdkMenuActivity.this, "登录成功 ",

Toast.LENGTH\_LONG).show();

}

@Override

public void onFailure(int code, String msg) {

Toast.makeText(AliSdkMenuActivity.this, "登录失败 ",

Toast.LENGTH\_LONG).show();

}

});

注销淘宝账号：

AlibcLogin alibcLogin = AlibcLogin.getInstance();

alibcLogin.logout(AliSdkMenuActivity.this, new LogoutCallback() {

@Override

public void onSuccess() {

Toast.makeText(AliSdkMenuActivity.this, "退出登录成功",

Toast.LENGTH\_SHORT).show();

}

@Override

public void onFailure(int code, String msg) {

Toast.makeText(AliSdkMenuActivity.this, "退出登录失败 " + code

+ msg,Toast.LENGTH\_SHORT).show();

}

});

### 打开商品详情页

通过百川SDK来完成。

使用方法：

private Map<String, String> exParams = new HashMap<>();

WebView webView = (WebView)findViewById（R.id.webView）;

webView.getSettings().setJavaScriptEnabled(true);

exParams.put(AlibcConstants.*ISV\_CODE*, "appisvcode");  
 exParams.put("alibaba", "阿里巴巴");//自定义参数部分，可任意增删改  
  
 //商品ID  
 String cid = getIntent().getStringExtra("num\_iid");  
 AlibcTaokeParams alibcTaokeParams = new AlibcTaokeParams("mm\_120715455\_0\_0", "mm\_120715455\_0\_0", null); // 若非淘客taokeParams设置为null即可  
// AlibcBasePage alibcBasePage = new AlibcDetailPage("532128520567");  
 AlibcBasePage alibcBasePage = new AlibcDetailPage(cid);  
 AlibcShowParams alibcShowParams = new AlibcShowParams(OpenType.*H5*, false);  
 AlibcTrade.*show*(AliSdkWebViewProxyActivity.this,webView, null, null,alibcBasePage,alibcShowParams,alibcTaokeParams,exParams, new DemoTradeCallback());

TradeCallBack类

public class DemoTradeCallback implements AlibcTradeCallback {  
  
 @Override  
 public void onTradeSuccess(TradeResult tradeResult) {  
 //当addCartPage加购成功和其他page支付成功的时候会回调  
  
 if(tradeResult.resultType.equals(ResultType.*TYPECART*)){  
 //加购成功  
 Toast.*makeText*(MyApplication.*application*, "加购成功", Toast.*LENGTH\_SHORT*).show();  
 LogUtil.*d*("加购成功");  
 }else if (tradeResult.resultType.equals(ResultType.*TYPEPAY*)){  
 //支付成功  
 Toast.*makeText*(MyApplication.*application*, "支付成功,成功订单号为"+tradeResult.payResult.paySuccessOrders, Toast.*LENGTH\_SHORT*).show();  
 LogUtil.*d*("支付成功,成功订单号为"+tradeResult.payResult.paySuccessOrders);  
 }  
 }  
  
 @Override  
 public void onFailure(int errCode, String errMsg) {  
 Toast.*makeText*(MyApplication.*application*, "电商SDK出错,错误码="+errCode+" / 错误消息="+errMsg, Toast.*LENGTH\_SHORT*).show();  
 LogUtil.*d*("电商SDK出错,错误码="+errCode+" / 错误消息="+errMsg);  
 }  
}

**说明：**打开商品详情页必须传入**阿里妈妈淘客PID**，以告知淘宝后台，用户是通过哪个淘宝客账户(也就是淘宝联盟账户)引导的，所产生的访问数据、订单数据以及淘宝佣金才会返回到企业的淘宝联盟账户里。

AlibcTaokeParams alibcTaokeParams = new AlibcTaokeParams(**阿里妈妈淘客PID**, "", null);

### 打开淘宝购物车

打开APP当前登录的淘宝账户的购物车。

使用方法：

**public** **void** showMyCartsPage(View view){

        TradeService tradeService = AlibabaSDK.getService(TradeService.**class**);

        MyCartsPage myCartsPage = **new** MyCartsPage();

                TradeConfigs.defaultISVCode = "kdkdkdkd" //传入isv\_code//即将

        tradeService.show(myCartsPage, **null**, MainActivity.**this**, **null**, **new** TradeProcessCallback(){

            @Override

**public** **void** onFailure(**int** code, String msg) {

                Toast.makeText(MainActivity.**this**, "失败 "+code+msg,

                        Toast.LENGTH\_SHORT).show();

              }

            @Override

**public** **void** onPaySuccess(TradeResult tradeResult) {

                Toast.makeText(MainActivity.**this**, "成功", Toast.LENGTH\_SHORT)

                        .show();

            }});

    }

### 打开淘宝订单

**public** **void** showMyOrdersPage(View view){

        TradeService tradeService = AlibabaSDK.getService(TradeService.**class**);

        MyOrdersPage myOrdersPage = **new** MyOrdersPage(0, **false**);

        tradeService.show(myOrdersPage, **null**, MainActivity.**this**, **null**, **new** TradeProcessCallback(){

            @Override

**public** **void** onFailure(**int** code, String msg) {

                Toast.makeText(MainActivity.**this**, "失败 "+code+msg,

                        Toast.LENGTH\_SHORT).show();

            }

            @Override

**public** **void** onPaySuccess(TradeResult tradeResult) {

                Toast.makeText(MainActivity.**this**, "成功", Toast.LENGTH\_SHORT)

                       .show();

            }});

    }

## 本地服务器接口

# 本地服务端接口

## 2.1.淘宝订单订阅消息

当通过淘宝联盟账户引导的买家创建的订单发生变化时，会接收到淘宝后台发送的订阅消息。

TmcClient client = **new** TmcClient(TOP\_KEY,APP\_SECRET, "default");// 请求地址sdk默认已经封装

client.setMessageHandler(**new** MessageHandler() {

**public** **void** onMessage(Message message, MessageStatus status) {

**try** {

System.*out*.println("============");

System.*out*.println(message.getContent());

System.*out*.println(message.getTopic());

// 默认不抛出异常则认为消息处理成功

} **catch** (Exception e) {

e.printStackTrace();

status.fail();// 消息处理失败回滚，服务端需要重发

}

}

});

**try** {

client.connect();

System.*out*.println("TMC是否在线 ：" + client.isOnline());

} **catch** (LinkException e) {

e.printStackTrace();

}

System.*out*.println(client.isOnline());

## 2.2.用户注册接口

|  |  |
| --- | --- |
| **承载协议：** | HTTP POST |
| **接口方向：** | Android客户端 –> 悟空返利后台 |
| **接口地址：** | host: http://192.168.0.4:8080/  api: WukongServer/user\_register |
| **接口功能:** | 用户注册 |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| userInfo | 用户基本信息 | String(json) | 必须 |
| userAuths | 用户认证信息 | String(json) | 必须 |

**userInfo结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| nick\_name | 用户名，示例：张三、zhangsan。。。 | String | 可选 |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int | 可选 |
| avatar | 头像，手机号码注册的用户，若修改头像，则必须以文件形式上传； | String | 可选 |
| balance | 余额， | double | 可选 |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double | 可选 |
| integral | 积分 | int | 可选 |
| invite\_code | 邀请码 | String | 可选 |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String | 必须 |

**userAuths结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String | 必须 |
| identifier | 账号 | String | 必须 |
| credential | 密码 | double | 必须 |

**请求方式：**

**请求地址 = host + api**，在代码中发起HTTP POST请求传入接口所需的参数即可。

**响应参数：**

该API返回一个Map<String,Object> json格式数据，开发者可对其进行解析，选取需要的信息。

|  |  |  |
| --- | --- | --- |
| **Map<String,Object>中数据** | **说明** | **数据**  **类型** |
| result | 响应结果，必定返回 | json |
| userInfo | 用户基本信息，成功返回 | json |
| userAuths | 用户认证信息，成功返回 | json |

**result结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| code | 响应码，200：成功，401：手机号已被注册 | int |
| message | 响应消息 | String |

**userInfo结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户在数据库中的唯一标识 | int |
| nick\_name | 用户名 | String |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int |
| avatar | 头像，格式为一个URL | String |
| balance | 余额， | double |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double |
| integral | 积分 | int |
| invite\_code | 邀请码 | String |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String |

**userAuths结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户认证信息在数据库中的唯一标识 | int |
| user\_id | 用户在数据库中的唯一标识 | int |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String |
| identifier | 账号 | String |
| credential | 密码 | double |

**响应示例：**



## 2.3.用户登录接口

|  |  |
| --- | --- |
| **承载协议：** | HTTP POST |
| **接口方向：** | Android客户端 –> 悟空返利后台 |
| **接口地址：** | api: WukongServer/user\_login |
| **接口功能:** | 用户登录 |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| userInfo | 用户基本信息，当使用三方账号登录时，必选传 | String(json) | 特殊可选 |
| userAuths | 用户认证信息 | String(json) | 必须 |
| loginInfo | 用户登录信息 | String(json) | 必须 |

**userInfo结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| nick\_name | 用户名，示例：张三、zhangsan。。。 | String | 可选 |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int | 可选 |
| avatar | 头像，手机号码注册的用户，若修改头像，则必须以文件形式上传； | String | 可选 |
| balance | 余额， | double | 可选 |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double | 可选 |
| integral | 积分 | int | 可选 |
| invite\_code | 邀请码 | String | 可选 |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String | 可选 |

**userAuths结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String | 必须 |
| identifier | 账号 | String | 必须 |
| credential | 密码 | String | 必须 |

**loginInfo结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| last\_login\_time | 登录时间，格式“2017-01-14 14:28:10” | String | 必须 |
| last\_login\_ip | 登录IP，示例：192.168.1.1 | String | 必须 |
| last\_login\_device\_model | 设备型号，示例：MI4 LTE | String | 必须 |
| last\_login\_device\_id | 设备ID，示例：4564564646431 | String | 必须 |

**请求方式：**

**请求地址 = host + api**，在代码中发起HTTP POST请求传入接口所需的参数即可。

**响应参数：**

该API返回一个Map<String,Object> json格式数据，开发者可对其进行解析，选取需要的信息。

|  |  |  |
| --- | --- | --- |
| **Map<String,Object>中数据** | **说明** | **数据**  **类型** |
| result | 响应结果，必定返回 | json |
| userInfo | 用户基本信息，成功返回 | json |
| userAuths | 用户认证信息，成功返回 | json |

**result结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| code | 响应码，200：成功，402：手机号还未注册，403：密码错误 | int |
| message | 响应消息 | String |

**userInfo结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户在数据库中的唯一标识 | int |
| nick\_name | 用户名 | String |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int |
| avatar | 头像，格式为一个URL，需拼接host地址才能正常使用 | String |
| balance | 余额， | double |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double |
| integral | 积分 | int |
| invite\_code | 邀请码 | String |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String |

**userAuths结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户认证信息在数据库中的唯一标识 | int |
| user\_id | 用户在数据库中的唯一标识 | int |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String |
| identifier | 账号 | String |
| credential | 密码 | String |

**响应示例：**



## 2.4.修改用户信息接口

|  |  |
| --- | --- |
| **承载协议：** | HTTP POST |
| **接口方向：** | Android客户端 –> 悟空返利后台 |
| **接口地址：** | api: WukongServer/user\_update |
| **接口功能:** | 修改用户信息，包括用户基本信息和修改用户登录密码 |

**请求参数：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| userInfo | 用户基本信息 | String(json) | 特殊可选 |
| userAuths | 用户认证信息，userInfo和userAuths必须传一个 | String(json) | 特殊可选 |
| avatar | 用户头像，当传此参数时，必须传userInfo参数 | File | 特殊可选 |

**userInfo结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| id | 用户在数据库中的唯一标识 | int | 必须 |
| nick\_name | 用户名，示例：张三、zhangsan。。。 | String | 必须 |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int | 必须 |
| avatar | 头像，手机号码注册的用户，若修改头像，则必须以文件形式上传； | String | 必须 |
| balance | 余额， | double | 必须 |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double | 可选 |
| integral | 积分 | int | 必须 |
| invite\_code | 邀请码 | String | 必须 |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String | 必须 |

**userAuths结构：**

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** | **是否必须** |
| id | 用户认证信息在数据库中的唯一标识 | int | 必须 |
| user\_id | 用户在数据库中的唯一标识 | int | 必须 |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String | 必须 |
| identifier | 账号 | String | 必须 |
| credential | 密码 | String | 必须 |

**avatar结构：**

|  |  |  |
| --- | --- | --- |
| **功能** | **说明** | **数据**  **类型** |
| 用户修改头像时，头像文件通过此参数上传 | Jpg/jpeg/png格式的图片，客户端需对图片进行压缩处理；详细上传方法见接口请求方式。 | File |

**请求方式：**

**请求地址 = host + api**，在代码中发起HTTP POST请求传入接口所需的参数即可。

**修改头像说明：**若用户修改了头像，则需在请求体中传入“avatar”参数，值为头像文件，支持JPG、JPEG、PNG图片格式，若没有修改则不要加“avatar”参数。传了“avatar”参数，则必须传“userInfo”参数。

请求示例：示例中使用了Android [OkHttp3](http://blog.csdn.net/itachi85/article/details/51190687)框架进行上传，仅供参考。示例中未对上传的图片进行压缩处理，开发使用时需自行处理。

//jpg或者jpeg格式图片  
private static final MediaType *MEDIA\_TYPE\_JPEG* = MediaType.*parse*("image/jpeg");  
//png格式图片  
private static final MediaType *MEDIA\_TYPE\_PNG* = MediaType.*parse*("image/png");

String rootPath = Environment.*getExternalStorageDirectory*().getAbsolutePath(); //手机根目录  
File headFile = new File(rootPath+"/images/head.jpg"); //需要上传的头像文件  
  
RequestBody requestBody = RequestBody.*create*(*MEDIA\_TYPE\_JPEG*,headFile);  
MultipartBody multipartBody = new MultipartBody.Builder()  
 .setType(MultipartBody.*FORM*)  
 .addFormDataPart("userInfo",userInfoJson) //用户基本信息  
 .addFormDataPart("avatar","1.jpg",requestBody) //上传后的文件名以用户ID命名，如:1.jpg  
 .build();  
  
Request request = new Request.Builder()  
 .url("http://192.168.0.4:8080/WukongServer/user\_update")  
 .post(multipartBody)  
 .build();  
  
new OkHttpClient().newCall(request).enqueue(new Callback() {  
 @Override  
 public void onFailure(Call call, IOException e) {  
 LogUtil.*d*("失败");  
 }  
  
 @Override  
 public void onResponse(Call call, Response response) throws IOException {  
 LogUtil.*d*("成功");  
 LogUtil.*d*(response.body().string());  
 }  
});

**响应参数：**

该API返回一个Map<String,Object> json格式数据，开发者可对其进行解析，选取需要的信息。

|  |  |  |
| --- | --- | --- |
| **Map<String,Object>中数据** | **说明** | **数据**  **类型** |
| result | 响应结果，必定返回 | json |
| userInfo | 用户基本信息，成功返回 | json |
| userAuths | 用户认证信息，成功返回 | json |

**result结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| code | 响应码，200：成功 | int |
| message | 响应消息 | String |

**userInfo结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户在数据库中的唯一标识 | int |
| nick\_name | 用户名 | String |
| sex | 性别，可选值：0，1,2。0：未设置，1：男，2：女 | int |
| avatar | 头像，格式为一个URL，需拼接host地址才能正常使用 | String |
| balance | 余额， | double |
| tmp\_balance | 临时余额，用户通过APP购物得到的返利金额，当用户确认收货完成订单后，该余额才会累积到balance中 | double |
| integral | 积分 | int |
| invite\_code | 邀请码 | String |
| regist\_time | 注册时间，格式“2017-01-14 14:28:10” | String |

**userAuths结构：**

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **数据**  **类型** |
| id | 用户认证信息在数据库中的唯一标识 | int |
| user\_id | 用户在数据库中的唯一标识 | int |
| identity\_type | 账号类型,暂时只有一种类型，手机号码注册：phone | String |
| identifier | 账号 | String |
| credential | 密码 | String |

**响应示例：**

