统一认证服务接口文档

版本：V0.2.9

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **时间** | **修订人** | **版本** | **修订内容** |
| 2015-4-14 |  | V0.1 | 起稿 |
| 2015-4-16 | 顺炽国  叶霄霄  朱斌 | V0.2 | 1. 公共参数放置到HTTP自定义头中 2. 增加客户端版本号上传 3. 去掉Email相关注册、找回密码操作 4. 增加帐号在不同平台的登录说明 5. 增加子帐号解绑功能 6. 修改验证码发送及模板配置接口 |
| 2015-5-19 | 顺炽国 | V0.2.1 | 1.完善HTTP响应码描述 |
| 2015-7-7 | 顺炽国 | V0.2.2 | 1.修复API地址，统一返回结果 |
| 2015-7-22 | 叶霄霄 | V0.2.3 | 1.完善统一返回错误码 |
| 2015-7-23 | 叶霄霄 | V0.2.4 | 1.修复部分API参数错误 |
| 2015-7-23 | 叶霄霄 | V0.2.5 | 1.增加用户基本信息具体字段的数据校验 |
| 2015-7-24 | 叶霄霄 | V0.2.6 | 1.修复部分API参数错误 |
| 2015-7-27 | 朱斌 | V0.2.7 | 1.修复授权和短信部分API参数错误 |
| 2015-7-28 | 叶霄霄 | V0.2.8 | 1.增加验证码使用说明 |
| 2015-7-29 | 叶霄霄 | V0.2.9 | 1. 增加通过账号获取基本信息接口 2. 修改注册时的返回码用户编码不合法为公共的错误码账号不合法 |
| 2015-8-14 | 顺炽国 | V0.3 | 1.登录接口增加Token过期时间参数 |

目录

[1. 约定 8](#_Toc426120669)

[1.1. 请求地址定义 8](#_Toc426120670)

[1.2. 提交方法 8](#_Toc426120671)

[1.3. HTTP自定义头信息 8](#_Toc426120672)

[1.4. HTTP请求HEADER 9](#_Toc426120673)

[1.5. 请求示例 9](#_Toc426120674)

[1.6. 返回值 10](#_Toc426120675)

[1.7. JSONP支持 12](#_Toc426120676)

[1.8. 错误码约定 12](#_Toc426120677)

[2. 用户接口 13](#_Toc426120678)

[2.1. 验证用户参数 13](#_Toc426120679)

[2.1.1. 功能说明 13](#_Toc426120680)

[2.1.2. 约定 13](#_Toc426120681)

[2.1.3. 接口定义 13](#_Toc426120682)

[2.1.4. 错误代码 14](#_Toc426120683)

[2.1.5. 返回值 14](#_Toc426120684)

[2.2. 获取验证码 14](#_Toc426120685)

[2.2.1. 功能说明 14](#_Toc426120686)

[2.2.2. 约定 14](#_Toc426120687)

[2.2.3. 接口定义 14](#_Toc426120688)

[2.2.4. 错误代码 14](#_Toc426120689)

[2.2.5. 返回值 14](#_Toc426120690)

[2.3. 校验验证码 15](#_Toc426120691)

[2.3.1. 功能说明 15](#_Toc426120692)

[2.3.2. 约定 15](#_Toc426120693)

[2.3.3. 接口定义 15](#_Toc426120694)

[2.3.4. 错误代码 15](#_Toc426120695)

[2.3.5. 返回值 15](#_Toc426120696)

[2.4. 用户注册 16](#_Toc426120697)

[2.4.1. 功能说明 16](#_Toc426120698)

[2.4.2. 约定 16](#_Toc426120699)

[2.4.3. 接口定义 16](#_Toc426120700)

[2.4.4. 错误代码 17](#_Toc426120701)

[2.4.5. 返回值 17](#_Toc426120702)

[2.5. 查询用户信息 18](#_Toc426120703)

[2.5.1. 功能说明 18](#_Toc426120704)

[2.5.2. 约定 18](#_Toc426120705)

[2.5.3. 接口定义 18](#_Toc426120706)

[2.5.4. 错误代码 18](#_Toc426120707)

[2.5.5. 返回值 18](#_Toc426120708)

[2.5.6. 示例 18](#_Toc426120709)

[2.6. 更新用户信息 19](#_Toc426120710)

[2.6.1. 功能说明 19](#_Toc426120711)

[2.6.2. 约定 19](#_Toc426120712)

[2.6.3. 接口定义 19](#_Toc426120713)

[2.6.4. 错误代码 19](#_Toc426120714)

[2.6.5. 返回值 19](#_Toc426120715)

[2.7. 修改用户密码 20](#_Toc426120716)

[2.7.1. 功能说明 20](#_Toc426120717)

[2.7.2. 约定 20](#_Toc426120718)

[2.7.3. 接口定义 20](#_Toc426120719)

[2.7.4. 错误代码 20](#_Toc426120720)

[2.7.5. 返回值 20](#_Toc426120721)

[2.8. 重新设置密码 20](#_Toc426120722)

[2.8.1. 功能说明 20](#_Toc426120723)

[2.8.2. 约定 21](#_Toc426120724)

[2.8.3. 接口定义 21](#_Toc426120725)

[2.8.4. 错误代码 21](#_Toc426120726)

[2.8.5. 返回值 21](#_Toc426120727)

[2.9. 删除用户（该版本未实现） 21](#_Toc426120728)

[2.9.1. 功能说明 21](#_Toc426120729)

[2.9.2. 约定 21](#_Toc426120730)

[2.9.3. 接口定义 22](#_Toc426120731)

[2.9.4. 错误代码 22](#_Toc426120732)

[2.9.5. 返回值 22](#_Toc426120733)

[2.10. 查询手机绑定状态 22](#_Toc426120734)

[2.10.1. 功能说明 22](#_Toc426120735)

[2.10.2. 约定 22](#_Toc426120736)

[2.10.3. 接口定义 22](#_Toc426120737)

[2.10.4. 错误代码 23](#_Toc426120738)

[2.10.5. 返回值 23](#_Toc426120739)

[2.11. 绑定手机 23](#_Toc426120740)

[2.11.1. 功能说明 23](#_Toc426120741)

[2.11.2. 约定 23](#_Toc426120742)

[2.11.3. 接口定义 23](#_Toc426120743)

[2.11.4. 错误代码 23](#_Toc426120744)

[2.11.5. 返回值 23](#_Toc426120745)

[2.12. 解绑手机 24](#_Toc426120746)

[2.12.1. 功能说明 24](#_Toc426120747)

[2.12.2. 约定 24](#_Toc426120748)

[2.12.3. 接口定义 24](#_Toc426120749)

[2.12.4. 错误代码 24](#_Toc426120750)

[2.12.5. 返回值 24](#_Toc426120751)

[2.13. 绑定应用账号 24](#_Toc426120752)

[2.13.1. 功能说明 24](#_Toc426120753)

[2.13.2. 约定 25](#_Toc426120754)

[2.13.3. 接口定义 25](#_Toc426120755)

[2.13.4. 错误代码 25](#_Toc426120756)

[2.13.5. 返回值 25](#_Toc426120757)

[2.14. 查询绑定账号 25](#_Toc426120758)

[2.14.1. 功能说明 25](#_Toc426120759)

[2.14.2. 约定 25](#_Toc426120760)

[2.14.3. 接口定义 26](#_Toc426120761)

[2.14.4. 错误代码 26](#_Toc426120762)

[2.14.5. 返回值 26](#_Toc426120763)

[2.15. 子账号解除绑定 26](#_Toc426120764)

[2.15.1. 功能说明 26](#_Toc426120765)

[2.15.2. 约定 26](#_Toc426120766)

[2.15.3. 接口定义 26](#_Toc426120767)

[2.15.4. 错误代码 26](#_Toc426120768)

[2.15.5. 返回值 27](#_Toc426120769)

[2.16. 禁用账号（管理员接口） 27](#_Toc426120770)

[2.16.1. 功能说明 27](#_Toc426120771)

[2.16.2. 约定 27](#_Toc426120772)

[2.16.3. 接口定义 27](#_Toc426120773)

[2.16.4. 错误代码 27](#_Toc426120774)

[2.16.5. 返回值 27](#_Toc426120775)

[2.17. 账号解禁（管理员接口） 28](#_Toc426120776)

[2.17.1. 功能说明 28](#_Toc426120777)

[2.17.2. 约定 28](#_Toc426120778)

[2.17.3. 接口定义 28](#_Toc426120779)

[2.17.4. 错误代码 28](#_Toc426120780)

[2.17.5. 返回值 28](#_Toc426120781)

[2.18. 查看用户状态（管理员接口） 28](#_Toc426120782)

[2.18.1. 功能说明 28](#_Toc426120783)

[2.18.2. 约定 29](#_Toc426120784)

[2.18.3. 接口定义 29](#_Toc426120785)

[2.18.4. 错误代码 29](#_Toc426120786)

[2.18.5. 返回值 29](#_Toc426120787)

[2.19. 通过账号查询用户信息 29](#_Toc426120788)

[2.19.1. 功能说明 29](#_Toc426120789)

[2.19.2. 约定 29](#_Toc426120790)

[2.19.3. 接口定义 29](#_Toc426120791)

[2.19.4. 错误代码 30](#_Toc426120792)

[2.19.5. 返回值 30](#_Toc426120793)

[3. 认证接口 30](#_Toc426120794)

[3.1. 认证接口 32](#_Toc426120795)

[3.1.1. 功能说明 32](#_Toc426120796)

[3.1.2. 约定 32](#_Toc426120797)

[3.1.3. 接口定义 33](#_Toc426120798)

[3.1.4. 错误代码 33](#_Toc426120799)

[3.1.5. 返回值 33](#_Toc426120800)

[3.1.6. 示例 34](#_Toc426120801)

[3.2. 使用UID登录 34](#_Toc426120802)

[3.2.1. 功能说明 34](#_Toc426120803)

[3.2.2. 约定 34](#_Toc426120804)

[3.2.3. 接口定义 34](#_Toc426120805)

[3.2.4. 错误代码 35](#_Toc426120806)

[3.2.5. 返回值 35](#_Toc426120807)

[3.3. 使用验证码登录 35](#_Toc426120808)

[3.3.1. 功能说明 35](#_Toc426120809)

[3.3.2. 约定 35](#_Toc426120810)

[3.3.3. 接口定义 35](#_Toc426120811)

[3.3.4. 错误代码 36](#_Toc426120812)

[3.3.5. 返回值 36](#_Toc426120813)

[3.4. 更新授权码 36](#_Toc426120814)

[3.4.1. 功能说明 36](#_Toc426120815)

[3.4.2. 约定 36](#_Toc426120816)

[3.4.3. 接口定义 36](#_Toc426120817)

[3.4.4. 错误代码 36](#_Toc426120818)

[3.4.5. 返回值 37](#_Toc426120819)

[3.5. 用户登录信息查询 37](#_Toc426120820)

[3.5.1. 功能说明 37](#_Toc426120821)

[3.5.2. 约定 37](#_Toc426120822)

[3.5.3. 接口定义 37](#_Toc426120823)

[3.5.4. 错误代码 37](#_Toc426120824)

[3.5.5. 返回值 38](#_Toc426120825)

[3.5.6. 示例 38](#_Toc426120826)

[3.6. 授权查询 38](#_Toc426120827)

[3.6.1. 功能说明 38](#_Toc426120828)

[3.6.2. 约定 38](#_Toc426120829)

[3.6.3. 接口定义 38](#_Toc426120830)

[3.6.4. 错误代码 39](#_Toc426120831)

[3.6.5. 返回值 39](#_Toc426120832)

[3.7. 注销授权 39](#_Toc426120833)

[3.7.1. 功能说明 39](#_Toc426120834)

[3.7.2. 约定 39](#_Toc426120835)

[3.7.3. 接口定义 39](#_Toc426120836)

[3.7.4. 错误代码 39](#_Toc426120837)

[3.7.5. 返回值 40](#_Toc426120838)

[4. 授权信息监听接口 40](#_Toc426120839)

[4.1. 用户授权推送注册 40](#_Toc426120840)

[4.1.1. 功能说明 40](#_Toc426120841)

[4.1.2. 约定 40](#_Toc426120842)

[4.1.3. 接口定义 40](#_Toc426120843)

[4.1.4. 返回值 41](#_Toc426120844)

[4.1.5. 错误码 41](#_Toc426120845)

[4.2. 用户授权推送解除 41](#_Toc426120846)

[4.2.1. 约定 41](#_Toc426120847)

[4.2.2. 接口定义 41](#_Toc426120848)

[4.2.3. 返回值 41](#_Toc426120849)

[4.2.4. 错误码 42](#_Toc426120850)

[4.3. 用户授权信息推送 42](#_Toc426120851)

[4.3.1. 推送内容 42](#_Toc426120852)

[5. 短信模板设置接口 43](#_Toc426120853)

[5.1.1. 功能说明 43](#_Toc426120854)

[5.1.2. 约定 43](#_Toc426120855)

[5.1.3. 接口定义 43](#_Toc426120856)

[5.1.4. 错误代码 44](#_Toc426120857)

[5.1.5. 返回值 44](#_Toc426120858)

# 约定

## 请求地址定义

接口建议采用HTTP协议，基于RESTFul标准，请求路径格式统一采用如下格式：

**http(s)**://**host:port**/**api/version**/**resource**?param1=xxxx&param2=xxxx……

说明：

1. http(s)根据服务器部署情况来确认是使用HTTP还是HTTPS协议
2. host:port指认证服务器的地址和服务端口号
3. api是固定名称
4. version是API的特定版本号，如:v1,v2等，目前版本号默认为V1
5. resource表示操作哪个资源如：Token（认证授权码）、SyncUser（用户同步）等
6. 参数可以使用URL的方式提交，也可以使用POST的方式

## 提交方法

**PUT**  更新数据

**DELETE** 删除数据

**GET**  查询数据

**POST**  新增或大量数据内容的更新、文件上传等

## HTTP自定义头信息

对于一些公共参数，如：客户端使用的协议版本号、应用ID、访问授权码（AccessToken）等，可以统一放置在HTTP自定义头中。

约定的HTTP自定义头如下：

|  |  |
| --- | --- |
| **HTTP头** | **说明** |
| XRK-CLIENT-VERSION | 客户端使用接口的版本号信息 |
| XRK-APPID | 客户端的AppID（注：此AppID是指应用的统一ID，与应用版本、平台无关） |
| XRK-ACCESS-TOKEN | 当前用户登录后得到的AccessToken，如果没有则不需要 |

目前约定的appId如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **AppName** | **AppID** | **应用名称** |
| **1** | substation | 1001 | 向日葵主站 |
| **2** | vip\_dashboard | 1002 | 我的向日葵（移动版） |
| **3** | bxr | 1003 | 微站 |
| **4** | iask | 1004 | 问吧 |
| **5** | community | 1005 | 社区 |
| **6** | sso\_web | 2001 | 统一认证站点 |
| **7** | xrk\_admin | 2002 | 向日葵后台管理系统 |
| **8** | koubei | 2003 | 口碑 |
| **9** | bxr\_admin | 2004 | 微站后台 |
| **10** | im\_api | 2005 | 消息API |
| **11** | xrk\_api | 2006 | 微站API |
| **12** | datahub | 2007 | 数据交换中心（Oracle） |
| **13** | account\_api | 2008 | 统一接口API |
| **14** | mysso | 2901 | 旧统一认证服务 |
| **15** | Weixin | 3001 | 微信 |

## HTTP请求HEADER

**User-Agent：**

统一认证服务只提供给内部应用使用，因此不考虑User-Agent头的格式。此信息交由对外服务接口控制。

**Content-Type:**

对于GET、DELETE请求，URL使用UrlEncode编码。

对于POST、PUT请求， Content-Type需要设置为：application/x-www-form-urlencoded

对于上传文件的请求则不在此接口文档讨论范围内。

注：请求的参数内容编码默认使用utf-8编码。

## 请求示例

GET /ACCOUNT/76885 HTTP/1.1

Connection: Keep-Alive

Host: api.xrk.com

POST /ACCOUNT HTTP/1.1

Connection: Keep-Alive

Content-Type: application/x-www-form-urlencoded;charset=UTF-8

Host: api.xrk.com

name=test&url=http%3A%2F%2www.xrk.com%2F13561234&mobile=13211223344&……

## 返回值

服务端返回默认返回为JSON格式，响应结果复用HTTP的状态码，因此客户端使用时应先判断HTTP的状态码再作二次判断，一般分为如下两种情况：

1. **执行成功**

此时返回内容默认为各个接口定义的返回对象，不作特殊说明

1. **执行失败**

默认的失败格式如下：

{

code: int, ---状态码，0默认为成功，其它见约定

message: string, --描述信息，如：异常时的一些描述

}

需要注意的是，HTTP响应码失败时一般会经过两次判断，第一次判断HTTP状态码属于哪种错误类型，再解析返回的详细错误响应内容。

附常见的操作返回的响应码

|  |  |  |
| --- | --- | --- |
| **方法** | **典型用法** | **返回状态码** |
| GET | - 获取表示  - 变更时获取表示（缓存） | 200（OK） - 表示已在响应中发出 |
| 204（无内容） - 资源有空表示 |
| 301（Moved Permanently） - 资源的URI已被更新 |
| 303（See Other） - 其他（如，负载均衡） |
| 304（not modified）- 资源未更改（缓存） |
| 400 （bad request）-指错误格式的请求 |
| 401 （unauthorized）-未授权的请求访问 |
| 403 （forbidden）-禁止访问指定的内容 |
| 404 （not found）- 资源不存在 |
| 405 (method not allowed) – 没有权限访问指定的方法 |
| 406 （not acceptable）- 服务端不支持请求的响应格式，如：请求的编码、语言、媒体格式、字符集等 |
| 415 （unsupported media type）-不支持请求的返回格式 |
| 422 （unprocessable entity）-不能处理的请求实体，一般用于检验参数失败的情况 |
| 500 （internal server error）- 通用错误响应 |
| 503 （Service Unavailable）- 服务端当前无法处理请求 |
| DELETE | - 删除资源 | 200 （OK）- 资源已被删除 |
| 301 （Moved Permanently）- 资源的URI已更改 |
| 303 （See Other）- 其他，如负载均衡 |
| 400 （bad request）-指错误格式的请求 |
| 401 （unauthorized）-未授权的请求访问 |
| 403 （forbidden）-禁止访问指定的内容 |
| 404 （not found）- 资源不存在 |
| 405 (method not allowed) – 没有权限访问指定的方法 |
| 409 （conflict）- 通用冲突 |
| 500 （internal server error）- 通用错误响应 |
| 503 （Service Unavailable）- 服务端当前无法处理请求 |
| PUT | - 用客户端管理的实例号创建一个资源  - 通过替换的方式更新资源  - 如果未被修改，则更新资源（乐观锁） | 200 （OK）- 如果已存在资源被更改 |
| 201 （created）- 如果新资源被创建 |
| 301（Moved Permanently）- 资源的URI已更改 |
| 303 （See Other）- 其他（如，负载均衡） |
| 400 （bad request）-指错误格式的请求 |
| 401 （unauthorized）-未授权的请求访问 |
| 403 （forbidden）-禁止访问指定的内容 |
| 404 （not found）- 资源不存在 |
| 406 （not acceptable）- 服务端不支持所需表示/p> |
| 405 (method not allowed) – 没有权限访问指定的方法 |
| 409 （conflict）- 通用冲突 |
| 412 （Precondition Failed）- 前置条件失败（主要用于并发控制，如多人同时修改时，根据HTTP头(If-Unmodified-Since或If-Match)判断资源是否允许修改） |
| 415 （unsupported media type）- 接受到的表示不受支持 |
| 422 （unprocessable entity）-不能处理的请求实体，一般用于检验参数失败的情况 |
| 500 （internal server error）- 通用错误响应 |
| 503 （Service Unavailable）- 服务当前无法处理请求 |
| POST | - 使用服务端管理的（自动产生）的实例号创建资源  - 创建子资源  - 部分更新资源  - 如果没有被修改，则不过更新资源（乐观锁） | 200（OK）- 如果现有资源已被更改 |
| 201（created）- 如果新资源被创建 |
| 202（accepted）- 已接受处理请求但尚未完成（异步处理） |
| 301（Moved Permanently）- 资源的URI被更新 |
| 303（See Other）- 其他（如，负载均衡） |
| 400（bad request）- 指错误格式的请求 |
| 401 （unauthorized）-未授权的请求访问 |
| 403 （forbidden）-禁止访问指定的内容 |
| 404 （not found）- 资源不存在 |
| 405 (method not allowed) – 没有权限访问指定的方法 |
| 406 （not acceptable）- 服务端不支持所需表示 |
| 409 （conflict）- 通用冲突 |
| 412 （Precondition Failed）- 前置条件失败（主要用于并发控制，如多人同时修改时，根据HTTP头(If-Unmodified-Since或If-Match)判断资源是否允许修改） |
| 415 （unsupported media type）-不支持请求的返回格式 |
| 422 （unprocessable entity）-不能处理的请求实体，一般用于检验参数失败的情况 |
| 500 （internal server error）- 通用错误响应 |
| 503 （Service Unavailable）- 服务当前无法处理请求 |

## JSONP支持

在需要跨域访问的情况下，认证服务接口提供对JSONP的支持功能，需要在请求参数中指定jsonp参数，例如：

<http://127.0.0.1/api/user/get?id=12&jsonp=callback_function_name>

那么返回的内容格式如下：

callback\_function\_name({

//用户信息对象

})

异常时则为：

callback\_function\_name({

code: 101,

message: “用户名必须输入”

})

**注：jsonp参数为保留参数，其它接口中不能使用此参数**

**说明：**

1. 如果无特殊说明，后须接口描述中默认只描述成功请求返回的内容，以及发生错误时的错误code。
2. 每个接口都会标名请求的resource，对应URL请求中的名称，调用者请自行组合。

## 错误码约定

Code一共8位，每两位为一组，其含义如下：

**10 10 21 10**

应用代码

模块代码

功能代码

异常代码

现有约定如下：

**应用：**

统一认证服务平台：10

**模块**

全局代码: 00

统一认证模块：01

用户管理：02

授权监听：03

**功能**

全局代码：00

其它功能同章节编号，如：2.1对应01，2.13对应13，以此类推。

公共异常代码列表（功能代码默认为00）：

10000001 accessToken已过期  
10000002 accessToken不存在

10000003 AppID不存在

10000004 时间戳无效

10000005 用户不存在

10000006 请求参数不正确

10000007 请求方法不正确

10000020 内部服务异常

10000021 验证码校验失败，不能继续当前操作

10000022 验证码校验结果已过期，不能继续当前操作

10000023 uid不合法

10000024 手机号码不合法

10000025 用户账号格式不合法

# 用户接口

## 验证用户参数

### 功能说明

注册时，验证用户的关键参数是否已被使用。

### 约定

**Resource** account/parameter

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| mobile | String | Yes | 手机号码 |

### 错误代码

### 返回值

{

mobile：boolean，-- true-可用，false-不可用

}

## 获取验证码

### 功能说明

注册、找回密码、绑定手机时，调用此接口获取手机验证码。

### 约定

**Resource** account/ captcha

**METHOD** PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| mobile | String | Yes | 手机号 |
| checkType | Int | Yes | 检查方式，1-注册，2-找回密码，3-绑定， 4-登录 |

### 错误代码

10020201 检查方式不合法

### 返回值

{

timeout:int 验证码过期时间(秒)

}

## 校验验证码

### 功能说明

校验注册验证码的有效性。

进行注册、修改密码、重置密码、绑定手机、绑定邮箱这几个操作之前，需要传入不同的检查方式(checkType)校验验证码，校验成功后才可以正常调用这几个接口。否则会请求失败。

### 约定

**Resource** account/ captcha

**METHOD** GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| mobile | String | Yes | 手机号码 |
| captcha | String | Yes | 用户产生的验证码 |
| checkType | int | Yes | 检查方式，1-注册，2-找回密码，3-绑定 |

### 错误代码

10020301 验证码不合法

10020302 验证码已过期

10020303 检查方式不合法

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 用户注册

### 功能说明

新增用户信息到统一认证服务中心，并获得统一认证中心所分配的用户唯一ID。

生成用户时，假如主账号时手机号，则自动绑定手机号；主账号为邮箱地址时同理。传入的用户基本信息中，自动过滤emailIsVerify、mobileIsVerify两个字段；假如主账号为手机号，自动过滤mobile字段；假如主账号为邮箱，自动过滤email字段。

### 约定

**Resource** account/user

**METHOD**  POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| mobile | String | Yes | 用户编码（一般用作对应App登录账号，目前默认为手机号） |
| password | String | Yes | 用户的密码，对原始密码经过MD5加密后的值 |
| userInfo | String | No | 用户的基础信息，使用JSON二次包装 |
| extendInfo | String | No | 用户的扩展信息，使用JSON二次包装 |
| unverified | String | NO | 注册时是否需要校验验证码。当且仅当该参数的值为true时，注册流程不需要校验验证码。 |

**注：**

userInfo数据结构如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必须提供 | 说明 |
| sex | int | No | 性别 |
| userName | String | No | 用户名 |
| mobile | String | No | 手机号 |
| email | String | No | 电子邮箱 |
| qq | String | No | 用户qq 号 |
| address | String | No | 地址 |
| postcode | String | No | 邮编 |

例：

{

sex:1,

userName:"张三",

mobile:"13300000000",

email:"xxx@xxx.com",

qq:"1222222222",

address:"广州市海珠区花城大道建滔广场",

postcode:"510006"

}

再序列化成字符串

extendInfo为用户的扩展信息，具体字段视用户场景而自定义，平台只负责存储，只需将相应键值对使用JSON包装传入即可。

如：

{

IDCode: "1111111111",

Site: "http://www.xxx.cc"

……

}

再序列化成字符串作为参数传入。

### 错误代码

10020401 密码不合法

10020402 用户基础信息不合法

10020403 用户扩展信息不合法

10020404 性别参数不合法

10020405 手机号码不合法

10020406 电子邮箱不合法

10020407 qq号码不合法

10020408 邮政编码不合法

10020499 账号已存在

### 返回值

{

uid: long, --生成的用户唯一ID

appId: int, --所属应用

userCode: String --用户编码

}

## 查询用户信息

### 功能说明

查询用户的基本信息。

### 约定

**Resource** account/user/{uid}

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |

注：{uid}替换为实际的用户ID。

### 错误代码

### 返回值

{

userInfo:Object 用户基本信息

extendInfo:Object 用户扩展信息

}

注：userInfo和extendInfo的说明参照[2.4.3](#_接口定义)

返回示例:

{

userInfo: {sex:1,userName:"张三",mobile:"13300000000"},

extendInfo: {IDCode:11111,site:"[www.aaa.cc](http://www.aaa.cc//)"}

}

### 示例

服务网站：[http://www.test.com](http://www.test.com/)

用户ID：10021

请求地址则为：

<http://www.test.com/account/user/10021>

## 更新用户信息

### 功能说明

更新用户的基本信息。用户的基本信息里，mobileIsVerify、emailIsVerify两个字段不允许修改，即使传入了也会自动过滤。当用户的主账号为手机号时，mobile字段不允许修改；当用户的主账号为邮箱账号时，email字段不允许修改，即使传入了也会自动过滤。

### 约定

**Resource** account/user/{uid}

**METHOD**  PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| userInfo | String | Yes | 用户的基础信息，使用JSON二次包装 |
| extendInfo | String | No | 用户的扩展信息，使用JSON二次包装 |

注：userInfo和extendInfo的说明参照[2.4.3](#_接口定义)

### 错误代码

10020601 用户基础信息不合法

10020602 用户扩展信息不合法

10020604 性别参数不合法

10020605 手机号码不合法

10020606 电子邮箱不合法

10020607 qq号码不合法

10020608 邮政编码不合法

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 修改用户密码

### 功能说明

更新用户密码。

### 约定

**Resource** account/password/{uid}

**METHOD**  PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| oldpwd | String | Yes | 原密码，对原始密码经过MD5加密后的值 |
| password | String | Yes | 新密码，对原始密码经过MD5加密后的值 |
| unverified | String | NO | 修改用户密码时，是否需要校验验证码。当且仅当该参数的值为true时，修改用户密码的流程不需要校验验证码。 |

### 错误代码

10020701 密码格式不合法

10020702 原始密码错误

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 重新设置密码

### 功能说明

使用找回密码功能的最后一步，重新设置用户密码。

### 约定

**Resource** account/password/{uid}

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| password | String | Yes | 新密码，对原始密码经过MD5加密后的值 |
| unverified | String | NO | 重置密码时，是否需要校验验证码。当且仅当该参数的值为true时，重置密码的流程不需要校验验证码。 |

### 错误代码

10020801 密码不合法

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 删除用户（该版本未实现）

### 功能说明

删除指定用户。

### 约定

**Resource** account/user/{uid}

**METHOD**  DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |

### 错误代码

10020901 没有操作权限

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

***注：作为安全考虑，删除用户需要考虑是否限制为只允许添加者删除。***

## 查询手机绑定状态

### 功能说明

查询账号是否已绑定手机。

### 约定

**Resource** account/bind/{uid}

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |

### 错误代码

### 返回值

{

mobile: boolean，--true-已绑定，false-未绑定

}

## 绑定手机

### 功能说明

账号绑定手机。用户注册时，假如主账号为手机号，则自动绑定手机。假如用户已经绑定了手机A，允许用户重新绑定手机B以替代手机A；在这种场景下，假如用户的主账号也为手机号A，则重新绑定手机B之后，用户的主账号也更换为手机B。

### 约定

**Resource** account/bind/{uid}

**METHOD**  PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| mobile | String | Yes | 假如是绑定手机，必须提供该字段 |
| unverified | String | NO | 绑定手机时，是否需要校验验证码。当且仅当该参数的值为true时，绑定手机的流程不需要校验验证码。 |

### 错误代码

10021102 手机号已被他人绑定

10021103 该用户已绑定手机

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 解绑手机

### 功能说明

将账号与手机号解绑。假如用户的主账号为手机号，则不允许解绑手机号。

### 约定

**Resource** account/bind/{uid}

**METHOD**  DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| unverified | String | NO | 解绑手机时，是否需要校验验证码。当且仅当该参数的值为true时，解绑手机的流程不需要校验验证码。 |

### 错误代码

10021200 该用户未绑定手机

10021201 不允许解绑手机（用户的主账号是手机）

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 绑定应用账号

### 功能说明

将已注册的用户绑定到未登录过的App上。比如用户正在使用公众号，此时若绑定用户的向日葵账号，则公众号可以获取向日葵账号的头像、昵称等信息用在公众号上。

### 约定

**Resource** account/subAccount/{uid}

**METHOD**  POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| tempId | String | Yes | 用户使用APP时生成的临时账号 |
| subAppId | int | Yes | 要绑定的子应用ID |

### 错误代码

10021301 tempID不合法

10021302 该子账号已被绑定

10021303 子应用ID不合法

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 查询绑定账号

### 功能说明

查询当前账号已关联的账号（包含第三方系统账号），需要注意的是，对于内部应用，返回的是与用户相关的临时ID。

### 约定

**Resource** account/subAccount/{uid}

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |

### 错误代码

### 返回值

[uidObj1, uidObj2, ……,uidObjn]

每个对象结构如下:

{

uid:String, --用户ID

bindAppId:int, --绑定账号所在的应用ID

bindAppName:String, --绑定账号所在的应用名称

account: String, --账号名称

appId:int, --账号对应的应用ID

appName:String --账号对应的应用名称

thridParty:Boolean --是否第三方应用，如果是则为true

}

## 查询指定子账号

### 功能说明

查询指定子账号在某个应用下的信息。

### 约定

**Resource** account/subAccount

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| **subaccount** | **String** | **Yes** | **子账号名称** |
| **subappid** | **Int** | **Yes** | **子账号所在的应用ID** |

### 错误代码

10000005 子账号不存在

### 返回值

{

uid:String, --用户ID

bindAppId:int, --绑定账号所在的应用ID

bindAppName:String, --绑定账号所在的应用名称

account: String, --账号名称

appId:int, --账号对应的应用ID

appName:String --账号对应的应用名称

thridParty:Boolean --是否第三方应用，如果是则为true

}

## 子账号解除绑定

### 功能说明

将对应app与已注册的账号解除绑定。

### 约定

**Resource** account/subAccount/{uid}

**METHOD**  DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| subAppId | int | Yes | 要解除绑定的应用ID |
| subAccount | String | Yes | 要解除绑定的子账号名称 |
| {uid} | Int | Yes | 绑定子账号的用户ID，通过URL参数传入 |

### 错误代码

10021501 子应用ID不合法

10021502 子账号不存在

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 禁用账号（管理员接口）

### 功能说明

封禁指定账号。这里的uid是指管理员账号的uid，而targetId指的是要操作的目标账号的uid。

### 约定

**Resource** account/status/{uid}

**METHOD**  PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| targetId | long | Yes | 目标用户的UID |

### 错误代码

10021601 用户已被禁用

10021699 没有操作权限

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 账号解禁（管理员接口）

### 功能说明

将指定账号解封。这里的uid是指管理员账号的uid，而targetId指的是要操作的目标账号的uid。

### 约定

**Resource** account/status/{uid}

**METHOD**  DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| targetId | long | Yes | 目标用户的UID |

### 错误代码

10021701 该用户未被禁用

10021799 没有操作权限

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

## 查看用户状态（管理员接口）

### 功能说明

查看用户是否被禁用。这里的uid是指管理员账号的uid，而targetId指的是要操作的目标账号的uid。

### 约定

**Resource** account/status/{uid}

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| targetId | long | Yes | 目标用户的UID |

### 错误代码

10021899 没有操作权限

### 返回值

{

result: Boolean, --布尔值, true-账号可用（未被禁用），false-账号不可用（被禁用）

}

## 通过账号查询用户信息

### 功能说明

通过用户账号，查询用户的基本信息。

### 约定

**Resource** account/user

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| account | String | Yes | 要查询的用户的账号 |

### 错误代码

### 返回值

{

userInfo:Object 用户基本信息

extendInfo:Object 用户扩展信息

}

注：userInfo和extendInfo的说明参照[2.4.3](#_接口定义)

返回示例:

{

userInfo: {sex:1,userName:"张三",mobile:"13300000000"},

extendInfo: {IDCode:11111,site:"[www.aaa.cc](http://www.aaa.cc//)"}

}

# 认证接口

移动端认证流程：



对于App端来说，用户是否共享，由

WEB端登录流程



对于Web来说，其实所有登录操作都是通过WEB登录服务进行的，如果某个Web应用不需要统一登录，则可以直接与统一认证服务对接。

## 认证接口

### 功能说明

提供登录认证的操作，同时取得接口访问授权 Key。同一个AppID只能允许存在一个登录会话。

### 约定

**Resource** AccessToken

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| Account | String | Yes | 用户账号，目前系统约定为手机号码 |
| password | String | Yes | 经过混淆（salt）加密后的密码，加密规则见下面的说明 |
| timestamp | Int | Yes | 当前的时间戳，基于Unix时间戳 |
| Scope | String | No | 请求授权的范围，保留参数 |
| expireTime | Int | No | 过期时间，如果不使用，则使用系统默认过期时间，时间单位为分钟 |

**参数说明**

**password:**

在数据库中存储的密码建议采用如下规则：

Md5(Md5(password)+account)

先对用密码进行md5运算得到的值与用户ID相加，再进行MD5计算，得到最终存储的密码。

对于上传的用户密码，是经过ticket混淆后的密码，其算法在原来数据库的基础上增加了一次md5运算，如下:

Md5(Md5(Md5(password)+account)+timestamp)

**timestamp:**

使用Unix的时间戳，从1970-1-1到现在的毫秒数，其作用主要是用于混淆及超时检测

### 错误代码

10010101 密码错误

### 返回值

{

uid: string, --对应用户ID

appId:string --对应的AppID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效，过期后不能使用

expireTime:datetime --过期时间

}

### 示例

UID： d16896080d144661b9150f2f705d177f

Password: 123456

Timestamp: 1427422210 (2015-3-27 10:10:10)

加密步骤如下：

1. 对原始密码进行md5运算：

md5(password) = e10adc3949ba59abbe56e057f20f883e

1. 将运算后的md5值与用户ID相加，进行第二次md5运算（此值也是存在数据库中的密码值）

md5(md5(password)+uid)

=md5(e10adc3949ba59abbe56e057f20f883ed16896080d144661b9150f2f705d177f)

= 924e6760ff0931dcb25a3aa8f072e42e

1. 对第二步计算后的值与当前时间戳相加进行第三次md5运算，得到最终的加密密码

Md5(Md5(Md5(password)+uid)+timestamp)

=md5(924e6760ff0931dcb25a3aa8f072e42e1427422210)

=4bf046e8c02d5098a72400baa633b50b

## 使用UID登录

### 功能说明

使用分配的UID登录系统。

### 约定

**Resource** AccessToken/{uid}

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| password | String | Yes | 仅对密码作md5运算 |
| timestamp | Int | Yes | 当前的时间戳，基于Unix时间戳 |
| Scope | String | No | 请求授权的范围，保留参数 |
| expireTime | Int | No | 过期时间，如果不使用，则使用系统默认过期时间，时间单位为分钟 |

### 错误代码

### 返回值

{

uid: string, --对应用户ID

appId:string --对应的AppID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效，过期后不能使用

expireTime:datetime --过期时间

}

## 使用验证码登录

### 功能说明

通过手机收到的验证码进行登录认证，同时取得接口访问授权 Key。同一个AppID只能允许存在一个登录会话，如果用户通过用户名和密码登录后，再通过验证码登录上次的授权码不会发生变化。

### 约定

**Resource** AccessToken/captcha

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| Mobile | String | Yes | 登录的手机号 |
| Captcha | String | Yes | 手机接收到的验证码 |
| Scope | String | No | 请求授权的范围，保留参数 |
| expireTime | Int | No | 过期时间，如果不使用，则使用系统默认过期时间，时间单位为分钟 |

**参数说明**

**Captcha:**

可以通过[获取验证码](#_获取验证码)接口中的checkType=4获取到。

### 错误代码

10020301 无效的验证码

### 返回值

{

uid: string, --对应用户ID

appId:string --对应的AppID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效，过期后不能使用

expireTime:datetime --过期时间

}

## 使用子账号免密登录

### 功能说明

如果用户绑定了某个应用下的子账号，那么可以使用子账号及关联的应用ID免密登录，获取到当前注册用户ID及登录的授权信息。

### 约定

**Resource** AccessToken/subaccount

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| subaccount | String | Yes | 子账号 |
| subAppId | String | Yes | 子账号所属的AppID |
| Scope | String | No | 请求授权的范围，保留参数 |
| expireTime | Int | No | 过期时间，如果不使用，则使用系统默认过期时间，时间单位为分钟 |

### 错误代码

10020301 无效的验证码

### 返回值

{

uid: string, --对应用户ID

appId:string --对应的AppID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效，过期后不能使用

expireTime:datetime --过期时间

}

## 更新授权码

### 功能说明

在授权码即将超时或需要重新获取授权码时，可以调用此接口重新获取得到新的授权码。

### 约定

**Resource** AccessToken

**METHOD**  PUT

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| uid | String | Yes | 当前登录用户的ID |
| refreshToken | String | Yes | 要获得的Token |
|  |  |  |  |

### 错误代码

10010201 refreshToken已过期

10010202 refreshToken不正确

### 返回值

{

uid: string, --对应用户ID

AppID: string --应用ID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效

expireTime:datetime --过期时间，使用unix timestamp格式

}

## 用户登录信息查询

### 功能说明

查询指定用户已登录的信息。

### 约定

**Resource** AccessToken/{uid}

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
|  |  |  |  |

注：{uid}替换为实际的用户ID

### 错误代码

10010301 用户未登录

### 返回值

[obj1, obj2……]

每个Obj的对象如下：

{

uid: string, --对应用户ID

AppID: string --应用ID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效

expireTime:datetime --过期时间，使用unix timestamp格式

}

### 示例

服务网站：[http://www.test.com](http://www.test.com/)

用户ID：10021

请求地址则为：

<http://www.test.com/accesstoken/10021>

## 授权查询

### 功能说明

验证授权码是否可用。

### 约定

**Resource** AccessToken

**METHOD**  GET

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
|  |  |  |  |
|  |  |  |  |

### 错误代码

### 返回值

{

uid: string, --对应用户ID

AppID: string --应用ID

accessToken: string, --访问Token

refreshToken: String, --获取新访问Token时使用，使用一次后失效

expireTime:datetime --过期时间，使用unix timestamp格式

}

## 注销授权

### 功能说明

注销已经授权的请求。

### 约定

**Resource** AccessToken

**METHOD**  DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
|  |  |  |  |
|  |  |  |  |

### 错误代码

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}

# 授权信息监听接口

## 用户授权推送注册

### 功能说明

在其它内部应用需要获取当前已授权用户的授权信息时，可以通过注册到推送接口，当有新用户授权成功时，将同步已授权的授权码信息到观察者提供的接口上。

注：

1. 推送的回调接口必须为有效的Http地址，且能正确接收POST请求
2. 推送的信息内容详见3.1.9章节定义的推送内容
3. 如果在不同网络环境，需要统一认证服务与观察者之间网段的访问问题

### 约定

**Resource** Authorize/push

**METHOD** POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| callBackUrl | String | Yes | 回调Url |

### 返回值

正常时的返回JSON数据包示例：

{

appID: string, --对应申请应用ID

status: string, -申请注册推送授权的状态,0表示成功,1表示失败。

regTime:datetime --注册时间

}

### 错误码

10000003 appID不存在

10030101 授权注册推送url地址不合法

10030102 此应用观察者已存在

## 用户授权推送解除

在其它内部应用不需要当前已授权用户的授权信息时，可以通过解除到推送接口，解除成功后已授权的授权码信息将不会推送到观察者提供的接口上。

### 约定

**Resource** Authorize/push

**METHOD** DELETE

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
|  |  |  |  |

### 返回值

正常时的返回JSON数据包示例：

{

appID: string, --对应申请应用ID

status: string, --解除注册推送授权的状态,0表示成功,1表示失败。

delTime:datetime --解除时间

}

### 错误码

10000003 appID不存在

10030103 观察者不存在

## 用户授权信息推送

因考虑到性能问题，统一认证服务推送不会采用实时推送的方式，而会根据不同的策略在缓冲区满或到指定推送时间后主动进行推送。

### 推送内容

以数组的格式推送到应用申请注册的回调url地址上

[authObj1, authObj2,……]

其中每个对象格式如下：

{

pushType: String, --消息类型，目前支持如下三种消息：

Authroize\_login，Authorize\_Update，Authroize\_Logout

message: --消息内容

{

uid: string, --对应用户ID

appId: String, --用户所属的应用ID

accessToken: string, --访问Token

refreshToken:string, --刷新Token

expireDate:datetime --过期时间，使用unix timestamp格式

}

注：

1. 观察者应用在进行授权码判断的时候，需要判断过期时间
2. 如果观察者应用未找到对应的授权码，可以调用3.1.4接口进行授权码验证查询，将获取到的结果放到本地应用的缓存中

## 短信模板设置接口

### 功能说明

不同的APP可针对用户注册、找回密码、等场景来设置发送短信的模板及短信模板的状态，如启用或禁用，来通过调用短信网关发送短信。

### 约定

**Resource** msg/template

**METHOD**  POST

### 接口定义

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名** | **类型** | **必须提供** | **说明** |
| appId | String | Yes | 所属应用 |
| msgInfo | String | Yes | 基础信息，使用JSON二次包装 |

**注：**

msgInfo数据结构如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 必须提供 | 说明 |
| templateCode | String | Yes | 模板业务code(01:注册,02:找回密码........) |
| content | String | Yes | 内容 |
| state | String | No | 状态0:启用,1:禁用,默认为启用, |

例：

{

templateCode:01,

content:"【向日葵】您在向日葵网的验证码:{code}",

state:"0",

}

再序列化成字符串

### 错误代码

10000003 appID不存在

10020500 模板信息不合法

### 返回值

{

result: Boolean, --布尔值, true-成功，false-失败

}