IDShells 接入开发手册

SafeShells

V1.0.2

目录

[第1章 IDShells接入指南 1](#_Toc518913868)

[1.1 准备工作 1](#_Toc518913869)

[第2章 业务接口使用说明 2](#_Toc518913870)

[2.1 初始化接口 2](#_Toc518913871)

[2.2 用户接口 2](#_Toc518913872)

[2.2.1 检查用户是否存在 2](#_Toc518913873)

[2.2.2 添加用户 3](#_Toc518913874)

[2.2.3 启用用户 3](#_Toc518913875)

[2.2.4 停用用户 4](#_Toc518913876)

[2.2.5 锁定用户 4](#_Toc518913877)

[2.2.6 解锁用户 5](#_Toc518913878)

[2.3 手机令牌(*Google* Authenticator) 5](#_Toc518913879)

[2.3.1 检查是否已经激活手机令牌 5](#_Toc518913880)

[2.3.2 获取手机令牌激活码 6](#_Toc518913881)

[2.3.3 OTP认证 7](#_Toc518913882)

[2.4 短信令牌 8](#_Toc518913883)

[2.4.1 发送短信口令 8](#_Toc518913884)

[2.5 邮件令牌 8](#_Toc518913885)

[2.5.1 发送邮件口令 8](#_Toc518913886)

[2.6 微信令牌激活（注册） 9](#_Toc518913887)

[2.6.1 检查是否已经激活微信令牌 9](#_Toc518913888)

[2.6.2 获取用户微信激活码 10](#_Toc518913889)

[2.6.3 激活微信令牌（微信服务器调用） 10](#_Toc518913890)

[2.7 微信令牌扫码认证 11](#_Toc518913891)

[2.7.1 获取认证二维码(第1步) 11](#_Toc518913892)

[2.7.2 获取微信扫码认证信息（微信服务器调用）(第2步) 12](#_Toc518913893)

[2.7.3 微信扫码认证响应（微信服务器调用）(第3步) 12](#_Toc518913894)

[2.7.4 扫码认证(第4步) 13](#_Toc518913895)

[第3章 返回码列表 15](#_Toc518913896)

# IDShells接入指南

## 准备工作

联系管理员获取AppKey（应用标识）、AppSecret（应用安全密钥）和 API服务器地址（应用服务器地址），并获取SDK开发包。

# 业务接口使用说明

## 初始化接口

接口功能：设置认证服务地址，appkey（应用标识）、appsecret（应用安全密钥）到开发包中，用于后续其它接口调用时提供访问令牌。(只初始化一次)

init(String serverUrl, String appKey, String appSecret)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| serverUrl | 认证服务地址 |
| appKey | 应用标识 |
| appSecret | 应用安全密钥,用于加密数据，计算MAC等 |

## 用户接口

### 检查用户是否存在

接口功能：在添加用户之前，可以先执行用户是否存在的检查，此接口主要用来判断用户是否已经存在。

String isUserExist(String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{"code":0, //0表示成功

"msg":"Message or Error Message",

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，201表示不存在，其它参见返回码列表 |
| msg | 返回消息或错误描述 |

### 添加用户

接口功能：添加用户

String addUser(String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

### 启用用户

接口功能：

String enableUser(String username)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

### 停用用户

接口功能：

String disableUser(String username)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |
|  |  |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

### 锁定用户

接口功能：

String lockUser(String username)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

### 解锁用户

接口功能：

String unLockUser(String username)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

## 手机令牌(Google Authenticator)

### 检查是否已经激活手机令牌

接口功能：检查用户是否已经激活手机令牌

String isMobileTokenActived(String username)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，204表示未激活，其它参见返回码列表 |
| msg | 返回消息或错误描述 |

### 获取手机令牌激活码

String getMobileTokenActiveData(String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"code": "0",

"msg": "成功",

"activecode":"otpauth://totp/userid?secret=",

"state": "原样返回"

}

属性说明：

|  |  |
| --- | --- |
| **属性名称** | **说明** |
| code | 返回码 |
| msg | 返回消息或错误描述 |
| activecode | 该属性值用于生成二维码，供支持Google动态令牌算法的APP扫描，以生成动态令牌。 |

### OTP认证

接口功能：使用用户名进行OTP认证，可认证手机令牌，短信令牌

String userOTPAuth(String username, String otp);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |
| otp | 用户输入的动态口令 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message", //错误消息说明

"retrycnt":1,

"authtoken":"6FB96FA0ED234079AAEDC6B3361A1F590860"

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |
| retrycnt | 当认证失败时，返回此认证重试次数 |
| authtoken | 此次认证成功后返回的唯一token |

## 短信令牌

### 发送短信口令

接口功能：使用用户名和手机号发送短信口令，用于认证，短信令牌的认证统一使用”OTP认证”接口。

String getSmsOtp(String username,String phoneSN)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |
| phoneSN | 手机号 |

返回结果：

{

"data":

{

"code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

## 邮件令牌

### 发送邮件口令

接口功能：使用用户名和手机号发送邮件口令，用于认证，邮件令牌的认证统一使用”OTP认证”接口。

String getEmailOtp(String username,String email)

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |
| email | 邮箱地址 |

返回结果：

{

"data":

{

"code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |

## 微信令牌激活（注册）

### 检查是否已经激活微信令牌

接口功能：检查用户是否已经激活微信令牌。

String isWechatTokenActived(String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，204表示未激活 |
| msg | 返回消息或错误描述，参见返回码列表 |

### 获取用户微信激活码

接口功能：获取用户微信授权码，用于微信激活，此激活码用于微信扫码后传递给微信服务器。

String getWechatTokenAC (String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message", //错误消息说明

"wechatac":"12345678"//微信激活码

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功 |
| msg | 返回消息或错误描述，参见返回码列表 |
| wechatac | 微信授权码 |

### 激活微信令牌（微信服务器调用）

接口功能：微信令牌激活（用户和微信令牌绑定）。

String activeWechatToken(String openid,String wechatAC);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| openid | 微信号 |
| wechatAC | 微信激活码 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message" //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功 |
| msg | 返回消息或错误描述，参见返回码列表 |

## 微信令牌扫码认证

微信扫码认证流程可按以下四步顺序调用。

### 获取认证二维码(第1步)

接口功能：扫描认证时，获取生成二维码的数据，调用完成展示二维码后，可立即调用第四步等待用户扫码。

String getQrcode (String username);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| username | 用户名,可以为空，用户名为空表示该企业下所有用户都可以进行扫描认证 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message"， //错误消息说明

"qrcodedata":"apihostname/reqid"，//二维码数据

"reqid":"reqid" //请求标示，云信令扫描认证接口参数

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功，参见返回码列表 |
| msg | 返回消息或错误描述 |
| qrcodedata | 二维码数据，生成二维码的数据 |
| reqid | 请求标示，云信令扫描认证接口参数 |

### 获取微信扫码认证信息（微信服务器调用）(第2步)

接口功能：获取扫码认证的详细信息。

String getWechatScanAuthInfo (String qrcode, String openid);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| qrcode | 二维码数据 |
| openid | 微信号 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message", //错误消息说明

"userid":"user1", //用户名

"appname":"测试应用", //应用名称

"reqid":" adc15028-bad8-4ded-8dc0-30d54f921058 "//请求唯一标示

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功 |
| msg | 返回消息或错误描述，参见返回码列表 |
| userid | 用户名 |
| appname | 应用名称 |
| reqid | 请求唯一标示 |

### 微信扫码认证响应（微信服务器调用）(第3步)

接口功能：微信扫描认证响应（确认登录、拒绝登录）。

String wechatScanAuthResponse (String reqid, String username, String result);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| reqid | 请求唯一标示 |
| username | 用户名 |
| result | 响应结果：0：拒绝登录 1：确认登录 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message", //错误消息说明

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功 |
| msg | 返回消息或错误描述，参见返回码列表 |

### 扫码认证(第4步)

接口功能：扫描认证，获取微信扫码认证结果

String scanAuth (String reqid);

参数说明：

|  |  |
| --- | --- |
| **参数** | **说明** |
| reqid | 请求标示，通过获取扫描登录二维码获取 |

返回结果：

{

"data":

{ "code":0, //0表示成功

"msg":"Message or Error Message", //错误消息说明

"authtoken":"6FB96FA0ED234079AAEDC6B3361A1F590860"

}

}

|  |  |
| --- | --- |
| 属性名称 | 说明 |
| code | 返回码，0表示成功 |
| msg | 返回消息或错误描述，参见返回码列表 |
| authtoken | 此次认证成功后返回的唯一token |

# 返回码列表

|  |  |
| --- | --- |
| 错误码 | 说明 |
| 0 | 业务成功响应 |
| **基础操作错误码** | |
| 1 | 业务请求参数缺少或参数类型不正确 |
| 2 | 处理请求失败，未知错误原因 |
| 3 | 处理数据解析错误，无效的请求 |
| 4 | 请求不支持，无效的请求 |
| 5 | 无效的客户端请求 |
| 6 | 数据解密失败，请检查安全密钥是否填写错误或已被服务器端更新 |
| 7 | 无权访问(接口) |
| OAuth2.0错误码 | |
| 101 | 无效应用标识，请检查是否已经添加应用或应用标识是否正确 |
| 102 | 无效的会话标识 |
| 103 | 会话标识已过期 |
| 104 | 无效的回调URI |
| 用户认证相关错误码 | |
| 201 | 用户不存在，请检查您输入的用户帐号是否正确 |
| 202 | 用户未启用 |
| 203 | 用户已锁定，请稍后重试 |
| 204 | 用户未绑定令牌 |
| 205 | 用户绑定令牌失败 |
| 206 | 用户没有应用的访问权限 |
| 207 | 手机号已存在 |
| 208 | 手机令牌激活码生成失败 |
| 209 | 用户已存在 |
| 211 | 用户未激活手机令牌 |
| 212 | 用户未激活微信令牌 |
| 306 | 解密令牌数据失败 |
| 401 | 动态口令验证失败 |
| 402 | 动态口令已经被使用过，请输入新的动态口令 |
| 403 | 手机令牌时间需要校准 |
| 407 | 时间校准失败 |
| 410 | 应用集成类型与代理类型不匹配，请检查应用类型选择是否正确 |
| 411 | 认证策略为：进行动态口令认证 |
| 412 | 拒绝认证 |
| 413 | 业务响应超时 |
| 416 | 扫码登录认证失败 |
| 417 | 用户账号不匹配 |
| 418 | 扫描认证请求已过期 |
| 421 | 没有可同步的令牌类型 |
| 422 | 认证处理中 |
| 424 | 客户端令牌己删除 |
| 426 | 无效认证策略，用户选择使用组策略，但不属于任何组 |
| 427 | 不支持此设备类型 |
| 441 | 企业已锁定 |
| 442 | 应用已锁定 |
| 443 | 用户数已达上限 |
| 701 | 此微信号已经被其他用户绑定 |
| 702 | 此用户已经开通了微信登录服务 |
| 704 | 微信激活码无效 |
|  | |