目录

接入说明

概述

本文是kscop API H5的部署文档，用于指导 kscop API H5的集成。

安装

1. 页面引用kscop.js
2. 初始化KSCOP对象

   字段说明： app_id 为客户在金山云后台申请的。审核成功后生效

```
var opKscop = new KSCOP({
    app_id: appId, // 客户在金山云后台申请的。审核成功后生效
    timeout: 3000, // 超时
    pre_init: true // 是否初始化时获取运营商参数
});
```

- 网络状态判断方法（非必选）

  说明：该方法可提升非纯4G网络环境下用户的使用体验，为可选方法，不使用该方法不影响正常接入使用。

```
// 判断网段
var net = opKscop.checkNetInfo();
// 如果是Wi-Fi提示 打开流量
if(net !== 'wifi') {
    // 调用网关
    opKscop.gateway({phone:1xxxxxxxxxx})
}else {
    // 不调用网关验证 弹出提示（可根据具体的业务场景）
}
```

- 调用网关接口

```
opKscop.gateway({phone: 1xxxxxxxxxx})
```

- 调用onGatewaySuccess、onGatewayFail 获取校验结果

```
// 网关调用成功后触发
opKscop.onGatewaySuccess(function(data){
    // 调用后端校验接口，获取是否是本机校验结果
    $.ajax({
        method: 'POST',
        url: '后端接口',
        data: '参数',
        success:function(data){
            // 校验是否是本机结果
        },
        error: function(data) {
            // 失败
        }
    })

})
// 网关调用失败后触发
.onGatewayFail(function(data){
    // 网关失败 弹出错误提示
})
```

- 请在head中添加代码

```
<meta content="always" name="referrer">
```

方法说明

gateway(options) 调用网关

参数options结构： { phone: ‘1xxxxxxxxxx’ }，传入需要校验的号码。

onGatewaySuccess(fn)网关成功返回

fn 成功返回函数，返回函数参数是Object, 结构：{ process_id: ‘xxxx’, phone:’1xxxxxxxxxx’, accesscode: ‘abc’ }。

onGatewayFail(fn) 网关失败返回

fn 失败返回函数,返回函数参数是Object, 结构：{ code: 100 }，错误码参考说明

checkNetInfo() 返回当前网络状态

返回cellular、wifi、unknown 三种状态。建议网络状态为cellular和unknown情况时调用网关接口，网络状态为wifi时不调

用网关接口。

错误码

错误码	说明	常见原因	解决方法
100	pre_gateway接口网络失败	断网，超时或者跨域	检测网络
101	pre_gateway接口返回数据错误	数据格式不对	查看文档确认格式正确
102	接口网络失败	断网，超时或者跨域	检测网络
103	电信接口失败	是否是数据网络	检测网络，服务查看日志
104	移动接口失败	是否是数据网络，备案refer是否与当前页面refer一致	检测网络，确认备案refer， 服务查看日志
105	record_token接口网络失败	断网，超时或者跨域	检测网络
106	record_token接口失败	是否是数据网络	检测网络，服务查看日志
107	联通接口失败	是否是数据网络	检测网络，服务查看日志
108	切换运营商接口失败	超时接口错误	检测网络,服务查看日志

接入说明

概述

本文是OnePass API 小程序的部署文档，用于指导 OnePass API 小程序的集成。

安装

1. 下载Demo
2. 初始化onepass对象

```
this.opInstance = new Onepass({
  app_id: '您申请的ID',
  timeout: 3000, // 超时时间，默认3000
  pre_init: true // 是否初始化时获取运营商参数；默认为true
})
```

3. 调用gateway方法，返回运营商地址

   说明： 可调用小程序 wx.getNetworkType进行网络判断后调用gateway方法；该方法可提升非纯4G网络环境下用户的使用体验，为可选方法，不使用该方法不影响正常接入使用；

```
this.opInstance.gateway('phone', function(err, url){
  if(!err){
    // 设置image控件的src属性
    that.setData({ operator_url: url })
  } else {
    // 失败，针对这种开发者可以接入短信，走短信通道
  }
})
```

4. wxml文件中放入一个image控件，image的图片地址是运营商地址

   说明： 用户如果有多次请求gateway接口的业务需要，请在调用gateway方法前将image里的路径清空；this.setData({ operator_url: '' })

```
<image style='width: 0; height:0' bindload='imgload' binderror='imgload' referrerPolicy="no-referrer" src="{{operator_url}}" wx:if="operator_url"></image>
```

5. 在image控件的load和error事件处理函数中调用onepass的getTokenStatus方法

```
this.opInstance.getTokenStatus(function(err, data){
if(!err){
  // token调用成功，调用check gateway接口，服务端校验是否成功

} else {
  // 失败，针对这种开发者可以接入短信，走短信通道
}
})
```

6. 前往小程序后台添加request合法域名

配置列表

服务器配置	地址
request合法域名	https://onepass.geetest.com
	https://id6.me
	https://www.cmpassport.com

https://opencloud.wostore.cn
https://nishub1.10010.com:38750

Request合法域名在小程序开发后台（开发>开发设置）中设置，配置时请注意将上方地址全部加入到列表里，配置成功后需要大约10分钟左右后生效（小程序规定）。

说明： 开发者在本地进行移动网络调试时，需要在真机调试模式下进行调试。

错误码

错误码	说明
100	gateway接口网络请求失败，网络超时或失败
101	gateway接口返回失败，检查app_id是否合法
102	电信接口网络请求失败，网络超时或失败
103	电信取号失败（检查是否开启4G，查看接口返回数据）
104	移动取号失败（检查是否开启4G，查看接口返回数据）
107	联通取号失败 （检查是否开启4G，查看接口返回数据）

iOS版一键登录SDK开发文档

iOS v2.3.4.7

概述

当前版本：2.3.4.7

下载SDK及相关文档

请在官网下载最新的SDK包（包含Demo+SDK+资源包） 点击下载 非开发人员想体验demo，可直接下载ipa包到iPhone手机 点击下载

一.准备工作

升级指南

手动集成所有版本通用方式：

替换SDK静态库： 删除旧版本SDK所有相关的`.framework`文件，清除缓存，再导入新版SDK中的所有`.framework` 文件

前置条件

- SDK支持Xcode 12+打包，iOS9.0+及以上版本。
- SDK支持中国移动、联通、电信4G的取号能力。
- SDK支持网络环境为

a.纯数据网络

b.数据网络与wifi网络双开

- 对于双卡手机，SDK取当前流量卡号

一键登录使用场景：

用户无需输入手机号码，只需集成并调用SDK拉起授权页方法，用户确认授权后，SDK会获取token，服务端携带token到运营商网关获取用户当前上网使用的流量卡号码，并返回给APP服务端。

本机号码校验(本机认证)使用场景：

用户通过SDK获取token，服务端携带手机号码和token去运营商网关进行校验比对，返回的校验结果为：用户当前流量卡号码与服务端携带的手机号码是否一致。

创建应用

提示：一个应用对应一个appid，多个应用（不同bundleID）需创建多个应用以对应多个appid

快速体验Demo

将创建应用时获得的AppID填入Demo工程中，修改工程BundleID为创建应用时绑定的BundleID即可

开发环境搭建

手动集成

1. **导入framework：** 将SDK压缩包中framework文件夹下所有资源添加到工程中（注意勾选Copy items if needed）

2. Xcode配置：

- **OtherLinkerFlags中 添加**-ObjC**：xcode->BuildSetting->Other Linker Flags 添加 **-ObjC**
- **添加libc++.1.tbd**: 在xcode->General->Linked Frameworks and Libraries中点击 **+** ，搜索并选择添加 **libc++.1.tbd**
- **如果需要支持ios12以下系统，请在build phases->lik binary with libaries中导入* *Network.framework** 依赖

二.SDK使用说明

1.初始化

方法原型

```
/**
初始化
@param appId 验证后台申请的appId
@param complete 预初始化回调block
*/
+(void)initWithAppId:(NSString *)appId complete:(nullable YjdlShComplete)complete;
```

接口作用

初始化SDK :传入用户的appId,请求服务接口以获取运营商配置信息,接口请求成功缓存相关信息

使用场景

- **在app启动时进行调用**
- 保证在预取号或一键登录前至少调用一次
- 只需调用一次，多次调用不会多次初始化，与一次调用效果一致

请求示例代码

1. 导入SDK头文件 #import \<YjdlShSDK/YjdlShSDK.h>
2. 在AppDelegate中的 didFinishLaunchingWithOptions方法中添加初始化代码

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
...
//初始化
[YjdlShSDKManager initWithAppId:APPID complete:\^(YjdlShCompleteResult * _Nonnull completeResult) {
}];
/**
* 建议在退出登录时再次调用初始化方法
*/
...
}
```

2.预取号

不建议 频繁的多次调用和在拉起授权页后调用

预取号方法回调中处理UI操作需手动切换到主线程

- **方法原型**

```
/**
* 预取号
* 此调用将有助于提高拉起授权页的速度和成功率
* 建议在一键登录前提前调用此方法，比如调一键登录的vc的viewdidload中、初始化成功的回调中
```

* 不建议在拉起授权页后调用

* 回调中如需UI操作，建议自行切到主线程

*/

+(void)preGetPhonenumber:(nullable YjdlShComplete)complete;

接口作用

电信、联通、移动预取号 :初始化成功后，如果当前为电信/联通/移动，将调用预取号，可以提前获知当前用户的手机网络环境是否符合一键登录的使用条件，成功后将得到用于一键登录使用的临时凭证，默认的凭证有效期60min(三大运营商一致)。

使用场景

- 建议在执行一键登录的方法前，提前一段时间调用此方法，比如调一键登录的vc的viewdidload中，或者rootVC的viewdidload中，或者app启动后，此调用将有助于提高拉起授权页的速度和成功率
- 不建议调用后立即调用拉起授权页方法（此方法是异步）
- 此方法需要1\~2s的时间取得临时凭证，因此也不建议和拉起授权页方法一起串行调用
- 不建议频繁的多次调用和在拉起授权页后调用
- **建议在判断当前用户属于未登录状态时使用，已登录状态用户请不要调用该方法**

请求示例代码

ObjC:

#import \<YjdlShSDK/YjdlShSDK.h>

//开发者调拉起授权页的vc

@implementation CustomLoginViewController

- (void)viewDidLoad {

[super viewDidLoad];

if (YourAppLoginStatus == NO) {

//预取号

[YjdlShSDKManager preGetPhonenumber:nil];

...

}

}

...

//拉起授权页

- (void)AuthPageLogin{

...

}

3.拉起授权页

在预取号成功后调用，预取号失败不建议调用。调用拉起授权页方法后将会调起运营商授权页面。该方法会拉起登录界面，**已登录状态请勿调用** 。

/**

一键登录拉起内置授权页&获取Token(区分拉起授权页之前和之后的回调)

@param yjdlShUIConfigure验证授权页参数配置

@param openLoginAuthListener 拉起授权页监听：拉起授权页面成功或失败的回调，拉起成功或失败均触发。当拉起失败时，oneKeyLoginListener不会触发。此回调的内部触发时机是viewDidAppear

@param oneKeyLoginListener 一键登录监听：拉起授权页成功后的后续操作回调，包括点击SDK内置的(非外部自定义)取消登录按钮，以及点击本机号码一键登录的回调。点击授权页自定义按钮不触发此回调

* 回调中如需UI操作，建议自行切到主线程

```
*/

+(void)quickAuthLoginWithConfigure:(YjdlSUIConfigure*)yjdlShUIConfigure

openLoginAuthListener:(YjdlShComplete)openLoginAuthListener oneKeyLoginListener:(YjdlShComplete)oneKeyLoginListener;
```

参数描述

参数	类型	说明
yjdlShUIConfigure**必填**	YjdlShUIConfigure	授权页控件属性配置对象
openLoginAuthListener **选填**	YjdlShComplete	拉起授权页的回调，拉起页面成功、失败均触发 **注意：在此回调中用户UI操作必须手动切换到主线程**
oneKeyLoginListener **必填**	YjdlShComplete	**一键登录回调，用于接收一键登录的结果，点一键登录成功、失败均触发，点自带的返回按钮也触发**

使用场景

- 用户进行一键登录操作时，调用一键登录方法，如果初始化成功，SDK将会拉起授权页面，用户授权后，SDK将返回取号 token给到应用客户端。
- 可以在多处调用
- 需在调用预初始化方法之后调用

一键登录逻辑说明

- 存在调用预初始化时获取的临时凭证，调用一键登录方法将立即拉起授权页面
- openLoginAuthListener： 拉起授权页面成功或失败的回调，拉起成功或失败均触发。当拉起失败时，oneKeyLoginListener不会触发。此回调的内部触发时机是viewDidAppear
- oneKeyLoginListener：一键登录监听， 拉起授权页成功后的后续操作回调，包括点击SDK内置的(非外部自定义)取消登录按钮，以及点击本机号码一键登录的回调。点击授权页自定义按钮不触发此回调
- 不存在临时凭证或临时凭证过期时(临时凭证有效期电信60min、联通60min、移动60min)，调用一键登录方法，将有一个很短的时延，待取号成功后拉起授权页面
- 取号失败时，返回失败

请求示例代码

ObjC:

1. 导入SDK头文件 #import \<YjdlShSDK/YjdlShSDK.h>
2. 在需要使用一键登录的地方调用一键登录接口

```
// 用户需要使用一键登录时的方法

- (void)quickLoginBtnClick:(UIButton *)sender {

__weak typeof(self) weakself = self;

CGFloat screenScale = [UIScreen mainScreen].bounds.size.width/375.0;

YjdlSUIConfigure* baseUIConfigure = [YjdlSUIConfigurenew];

baseUIConfigure.viewController = self;

baseUIConfigure.yjdlShLogoImage = [UIImage imageNamed:@"your_app_logo_image"];

baseUIConfigure.yjdlShAppPrivacyFirst = @[@"测试连接1",[NSURL URLWithString:@"https://baidu.com"]];

NSString *filePath = [[NSBundle mainBundle] pathForResource:@"Index" ofType:@"html"];

baseUIConfigure.yjdlShAppPrivacySecond = @[@"本地协议地址",[NSURL fileURLWithPath:filePath]];

//layout 布局

...

// show loading...

//一键登录接口（将拉起授权页）

[YjdlShSDKManager quickAuthLoginWithConfigure:baseUIConfigure openLoginAuthListener:\^(YjdlShComplete * _Nonnull completeResult) {
```

```
if (completeResult.error) {
//切换到其他登录方式
}else{
}
} oneKeyLoginListener:\^( YjdlShComplete * _Nonnull completeResult) {
CFAbsoluteTime end = CFAbsoluteTimeGetCurrent();
__strong typeof(self) strongSelf = weakSelf;
if (completeResult.error) {
NSLog(@"oneKeyLoginListener:%@\ncost:%f",completeResult.yy_modelToJSONObject,end - start);
//提示：错误无需提示给用户，可以在用户无感知的状态下直接切换登录方式
if (completeResult.code == 1011){
//用户取消登录（点返回）
//处理建议：如无特殊需求可不做处理，仅作为交互状态回调，此时已经回到当前用户自己的页面
//点击sdk自带的返回，无论是否设置手动销毁，授权页面都会强制关闭
} else{
//处理建议：其他错误代码表示验证通道无法继续，可以统一走开发者自己的其他登录方式，也可以对不同的错误单独处理
//1003 一键登录获取token失败
//其他 其他错误//
//关闭授权页
// [YjdlShSDKManager finishAuthControllerAnimated:YES Completion:nil];
dispatch_async(dispatch_get_main_queue(), \^{
[YjdlShSDKManager hideLoading];
});
}
}else{
//置换手机号
}
}];
```

返回参数示例 completeResult.data

```
{
token = "A3-KqvM4yO_lApRLwpv4qVNnBRF3w=="
}
```

字段	类型	含义
token	String	token，置换令牌，用来和后台置换手机号。一次有效，有效期3min

- 用户取消登录(授权页点击返回) 【处理建议：若无特殊需求可不做处理】
- 用户选择其他方式登录(点击授权页自带的其他方式登录)：【处理建议：可根据实际情况跳转其他登录方式 】
- 其他错误 【处理建议：使用通道失败，可根据实际情况跳转其他登录方式 】

授权页销毁

注：sdk拉起授权页成功后，只允许点击一次一键登录按钮。一键登录按钮点击一次后，只有sdk自带的左上角返回按钮可以交互，效果为强制关闭授权页，其他页面元素均会被禁用，以防止多次点击导致多次回调而出现异常。

拉起授权页前 配置授权页面销毁机制属性

```
// 获取默认参数配置
YjdlSUIConfigure* baseUIConfigure = [YjdlSUIConfigureyjdlShDefaultUIConfigure];
baseUIConfigure.viewController = self;
// 是否需要手动销毁授权页面，默认自动销毁， YES->手动销毁
baseUIConfigure.manualDismiss = @(YES);
sdk获取token回调后销毁界面
//关闭页面
[YjdlShSDKManager finishAuthControllerCompletion:\^{
//如需关闭后present/push新页面，建议在completion回调中执行
//注：若未拉起授权页，调用此方法，block不会触发
//用户跳转短信验证
CustomSmsViewController * smsVc = [[CustomSmsViewController alloc]init];
CustomNavigationController * smsNav = [[CustomNavigationController alloc]initWithRootViewController:smsVc];
smsVc.navigationItem.title = @"短信验证";
[self presentViewController:smsNav animated:YES completion:nil];
}];
```

4.手动关闭授权页

当开发者设置点击一键登录或者自定义控件不自动销毁授权页时，将需要自行调用此方法主动销毁授权页，建议在置换手机号成功后销毁。如在得到回调后未销毁授权页而，使用拉起授权页方法再次拉起授权页，此页面将无法响应任何按键（除了导航栏的返回按钮）。

- **关闭授权页时机**

a.SDK拉起授权页方法 直接回调失败时

b.置换手机号有返回结果时

- **当前页面直接销毁**

```
//方式1
[self.PresentedViewController dismissViewControllerAnimated:YES completion:nil];
//方式二
[YjdlShSDKManager finishAuthControllerCompletion:\^{
//如需关闭后present/push新页面，建议在completion回调中执行
//注：若未拉起授权页，调用此方法，block不会触发
//用户跳转短信验证
CustomSmsViewController * smsVc = [[CustomSmsViewController alloc]init];
CustomNavigationController * smsNav = [[CustomNavigationController alloc]initWithRootViewController:smsVc];
smsVc.navigationItem.title = @"短信验证";
[self presentViewController:smsNav animated:YES completion:nil];
}];
```

- **找到topVC进行dismiss**

```
dispatch_async(dispatch_get_main_queue(), \^{
//建议使用授权页面配置对象传入的viewcontroller 调 dismiss
if (self.navigationController.viewControllers.lastObject.navigationController) {
[self.navigationController.viewControllers.lastObject dismissViewControllerAnimated:YES completion:nil];
} else {
```

```
UIViewController *topRootViewController = [[UIApplication sharedApplication] keyWindow].rootViewController;
// 在这里加一个这个样式的循环
while (topRootViewController.presentedViewController) {
// 这里固定写法
topRootViewController = topRootViewController.presentedViewController;
}
// 然后再进行present操作
[topRootViewController dismissViewControllerAnimated:YES completion:nil];
}
});
```

5.登录按钮、协议、勾选框控件点击监听

控件点击的监听是通过代理实现，在授权页面显示前触发代理进行监听。不监听协议点击可不实现代理，按需配置。

注意实现协议：**@interface xxxController ()\<YjdlShSDKManagerDelegate>**

在拉起授权页之前设置代理

```
//弹窗添加蒙版或点击协议回调 代理设置,不监听协议点击或不使用蒙版可不设置代理
[YjdlShSDKManager setYjdlShSDKManagerDelegate:self];
```

一键登录按钮、协议、勾选框点击监听

```
/**
* 统一事件监听方法
* type：事件类型（1，2，3）
* 1：隐私协议点击
* - 同-yjdlShSDKManagerWebPrivacyClicked:privacyIndex:currentTelecom
* code：0,1,2,3（协议页序号），message：协议名_当前运营商类型
* 2：协议勾选框点击
* code：0,1（0为未选中，1为选中）
* 3："一键登录"按钮点击
* code：0,1（0为协议勾选框未选中，1为选中）
*/
-(void) yjdlShActionListener:(NSInteger)type code:(NSInteger)code message:(NSString *_Nullable)message;
```

6.多次点击一键登录按钮、手动关闭授权页面上的loading

使用场景：授权页面“一键登录”按钮点击后oneKeyLoginListener失败回调中或用户APP端向服务端置换手机号码接口失败回调中调用该方法，以达到隐藏loading 再次点击"一键登录"按钮的效果(按钮最多只能被点击4次)。

```
// 隐藏授权页面上的loading框
+(void)hideLoading;
```

三.授权界面修改

1.设计规范

开发者不得通过任何技术手段，将授权页面的隐私栏、品牌露出内容隐藏、覆盖，对于接入SDK并上线的应用，我方和运营商会对上线的应用授权页面做审查，如果有出现未按要求设计授权页面，将隐私栏、运营商品牌、授权登录按钮隐去不可见的设计，我方有权将应用的登录功能下线。

2.页面可调整属性

注：授权页基本控件均支持上、下、左、右、宽、高、水平中心、竖直中心布局设置，布局通过布局对象设置，布局定位更加方

便快捷，建议使用最新布局对象进行设置。每个控件对象需要设置4个方向上的数值。

```
@class YjdlShUIConfigure;
/*
注： 授权页一键登录按钮、运营商品牌标签、运营商条款必须显示，不得隐藏，否则取号能力可能被运营商关闭
**/
//授权页UI配置
@interface YjdlShUIConfigure: NSObject
//要拉起授权页的vc [必填项] (注：SDK不持有接入方VC)
@property (nonatomic,weak)UIViewController * viewController;
/**
*外部手动管理关闭界面
*BOOL,default is NO
*eg.@(YES)
*/
@property (nonatomic,strong)NSNumber * manualDismiss;
/**授权页-背景图片*/
@property (nonatomic,strong) UIImage *yjdlShBackgroundImg;
/**授权页-背景色*/
@property (nonatomic,strong) UIColor *yjdlShBackgroundColor;
//导航栏
/**导航栏 是否隐藏 BOOL default is NO, 设置优先级高于yjdlShNavigationBackgroundClear eg.@(NO)*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBarHidden;
/**导航栏 背景透明 BOOL eg.@(YES)*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBackgroundClear;
/**导航栏标题*/
@property (nonatomic,strong)NSAttributedString * yjdlShNavigationAttributesTitleText;
/**导航栏右侧自定义按钮*/
@property (nonatomic,strong)UIBarButtonItem * yjdlShNavigationRightControl;
/**导航栏左侧自定义按钮*/
@property (nonatomic,strong)UIBarButtonItem * yjdlShNavigationLeftControl;
// 返回按钮
/**导航栏左侧返回按钮图片*/
@property (nonatomic,strong)UIImage * yjdlShNavigationBackBtnImage;
/**导航栏自带返回按钮隐藏，默认显示 BOOL eg.@(YES)*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBackBtnHidden;
/**************新增*****************/
/**返回按钮图片缩进 btn.imageInsets = UIEdgeInsetsMake(0, 0, 20, 20)*/
@property (nonatomic,strong)NSValue * yjdlShNavBackBtnImageInsets;
/**自带返回(关闭)按钮位置 默认NO 居左,设置为YES居右显示*/
@property (nonatomic,strong)NSNumber * yjdlShNavBackBtnAlimentRight;
```

```
/**隐藏导航栏分割线*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBottomLineHidden;
/**导航栏 文字颜色*/
@property (nonatomic,strong)UIColor * yjdlShNavigationTintColor;
/**导航栏 背景色 default is white*/
@property (nonatomic,strong)UIColor * yjdlShNavigationBarTintColor;
/**导航栏 背景图片*/
@property (nonatomic,strong)UIImage * yjdlShNavigationBackgroundImage;
/**导航栏 配合背景图片设置，用来控制在不同状态下导航栏的显示(横竖屏是否显示) UIBarMetrics eg.@(UIBarMetricsCompact)*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBarMetrics;
/**导航栏 导航栏底部分割线（图片）*/
@property (nonatomic,strong)UIImage * yjdlShNavigationShadowImage;
/*状态栏样式
*Info.plist: View controller-based status bar appearance = YES
*
*UIStatusBarStyleDefault：状态栏显示 黑
*UIStatusBarStyleLightContent：状态栏显示 白
*UIStatusBarStyleDarkContent：状态栏显示 黑 API_AVAILABLE(ios(13.0)) = 3
**eg. @(UIStatusBarStyleLightContent)
*/
@property (nonatomic,strong)NSNumber * yjdlShPreferredStatusBarStyle;
/*状态栏隐藏 eg.@(NO)*/
@property (nonatomic,strong)NSNumber * yjdlShPrefersStatusBarHidden;
/**
*NavigationBar.barStyle：默认UIBarStyleBlack
*Info.plist: View controller-based status bar appearance = YES
*UIBarStyleDefault：状态栏显示 黑
*UIBarStyleBlack：状态栏显示 白
*
*eg. @(UIBarStyleBlack)
*/
@property (nonatomic,strong)NSNumber * yjdlShNavigationBarStyle;
//LOGO图片
/**LOGO图片*/
@property (nonatomic,strong)UIImage * yjdlShLogoImage;
/**LOGO圆角 CGFloat eg.@(2.0)*/
@property (nonatomic,strong)NSNumber * yjdlShLogoCornerRadius;
/**LOGO显隐 BOOL eg.@(NO)*/
@property (nonatomic,strong)NSNumber * yjdlShLogoHiden;
/**手机号显示控件*/
```

```
/**手机号颜色*/
@property (nonatomic,strong)UIColor * yjdlShPhoneNumberColor;
/**手机号字体*/
@property (nonatomic,strong)UIFont * yjdlShPhoneNumberFont;
/**手机号对齐方式 NSTextAlignment eg.@(NSTextAlignmentCenter)*/
@property (nonatomic,strong)NSNumber * yjdlShPhoneNumberTextAlignment;
/*一键登录按钮 控件
注： 一键登录授权按钮 不得隐藏
**/
/**按钮文字*/
@property (nonatomic,copy)NSString * yjdlShLoginBtnText;
/**按钮文字颜色*/
@property (nonatomic,strong)UIColor * yjdlShLoginBtnTextColor;
/**按钮背景颜色*/
@property (nonatomic,strong)UIColor * yjdlShLoginBtnBgColor;
/**按钮文字字体*/
@property (nonatomic,strong)UIFont * yjdlShLoginBtnTextFont;
/**按钮背景图片*/
@property (nonatomic,strong)UIImage * yjdlShLoginBtnNormalBgImage;
/**按钮背景高亮图片*/
@property (nonatomic,strong)UIImage * yjdlShLoginBtnHightLightBgImage;
/**按钮背景不可用图片*/
@property (nonatomic,strong)UIImage * yjdlShLoginBtnDisabledBgImage;
/**按钮边框颜色*/
@property (nonatomic,strong)UIColor * yjdlShLoginBtnBorderColor;
/**按钮圆角 CGFloat eg.@(5)*/
@property (nonatomic,strong)NSNumber * yjdlShLoginBtnCornerRadius;
/**按钮边框 CGFloat eg.@(2.0)*/
@property (nonatomic,strong)NSNumber * yjdlShLoginBtnBorderWidth;
/*隐私条款Privacy
注： 运营商隐私条款 不得隐藏
用户条款不限制
**/
/**隐私条款 下划线设置，默认隐藏，设置yjdlShPrivacyShowUnderline = @(YES)显示下划线*/
@property (nonatomic,strong)NSNumber * yjdlShPrivacyShowUnderline;
/**隐私条款名称颜色：@[基础文字颜色UIColor*,条款颜色UIColor*] eg.@[[UIColor lightGrayColor],[UIColor greenColor]]*/
@property (nonatomic,strong) NSArray\<UIColor*> *yjdlShAppPrivacyColor;
/**隐私条款文字字体*/
@property (nonatomic,strong)UIFont * yjdlShAppPrivacyTextFont;
/**隐私条款文字对齐方式 NSTextAlignment eg.@(NSTextAlignmentCenter)*/
```

```
@property (nonatomic,strong)NSNumber * yjdlShAppPrivacyTextAlignment;

/**运营商隐私条款书名号 默认NO 不显示 BOOL eg.@(YES)*/

@property (nonatomic,strong)NSNumber * yjdlShAppPrivacyPunctuationMarks;

/**多行时行距 CGFloat eg.@(2.0)*/

@property (nonatomic,strong)NSNumber* yjdlShAppPrivacyLineSpacing;

/**是否需要sizeToFit,设置后与宽高约束的冲突请自行考虑 BOOL eg.@(YES)*/

@property (nonatomic,strong)NSNumber* yjdlShAppPrivacyNeedSizeToFit;

/**UITextView.textContainerInset 文字与TextView控件内边距 UIEdgeInset eg.[NSValue
valueWithUIEdgeInsets:UIEdgeInsetsMake(2, 2, 2, 2)]*/

@property (nonatomic,strong)NSValue* yjdlShAppPrivacyTextContainerInset;

/**隐私条款--APP名称简写 默认取CFBundledisplayname 设置描述文本四后此属性无效*/

@property (nonatomic,copy) NSString * yjdlShAppPrivacyAbbreviatedName;

/*

*隐私条款Y一:需同时设置Name和UrlString eg.@[@"条款一名称",条款一URL]

*@[NSSting,NSURL];

*/

@property (nonatomic,strong)NSArray * yjdlShAppPrivacyFirst;

/*

*隐私条款二:需同时设置Name和UrlString eg.@[@"条款一名称",条款一URL]

*@[NSSting,NSURL];

*/

@property (nonatomic,strong)NSArray * yjdlShAppPrivacySecond;

/*

*隐私条款三:需同时设置Name和UrlString eg.@[@"条款一名称",条款一URL]

*@[NSSting,NSURL];

*/

@property (nonatomic,strong)NSArray * yjdlShAppPrivacyThird;

/*

隐私协议文本拼接: DesTextFirst+运营商条款+DesTextSecond+隐私条款一+DesTextThird+隐私条款二+DesTextFourth+隐私条款三
+DesTextLast

**/

/**描述文本 首部 default:"同意"*/

@property (nonatomic,copy)NSString *yjdlShAppPrivacyNormalDesTextFirst;

/**描述文本二 default:"和"*/

@property (nonatomic,copy)NSString *yjdlShAppPrivacyNormalDesTextSecond;

/**描述文本三 default:"、"*/

@property (nonatomic,copy)NSString *yjdlShAppPrivacyNormalDesTextThird;

/**描述文本四 default:"、"*/

@property (nonatomic,copy)NSString *yjdlShAppPrivacyNormalDesTextFourth;

/**描述文本 尾部 default: "并授权AppName使用认证服务"*/

@property (nonatomic,copy)NSString *yjdlShAppPrivacyNormalDesTextLast;
```

```
/**运营商协议后置 默认@(NO)"*/
@property (nonatomic,strong)NSNumber *yjdlShOperatorPrivacyAtLast;
/**用户隐私协议WEB页面导航栏标题 默认显示用户条款名称*/
@property (nonatomic,strong)NSAttributedString * yjdlShAppPrivacyWebAttributesTitle;
/**运营商隐私协议WEB页面导航栏标题 默认显示运营商条款名称*/
@property (nonatomic,strong)NSAttributedString * yjdlShAppPrivacyWebNormalAttributesTitle;
/**自定义协议标题-按自定义协议对应顺序*/
@property (nonatomic,strong)NSArray\<NSString*> * yjdlShAppPrivacyWebTitleList;
/**隐私协议标题文本属性（用户协议&&运营商协议）*/
@property (nonatomic,strong)NSDictionary * yjdlShAppPrivacyWebAttributes;
/**隐私协议WEB页面导航返回按钮图片*/
@property (nonatomic,strong)UIImage * yjdlShAppPrivacyWebBackBtnImage;
/*协议页状态栏样式 默认：UIStatusBarStyleDefault*/
@property (nonatomic,strong)NSNumber * yjdlShAppPrivacyWebPreferredStatusBarStyle;
/**UINavigationTintColor*/
@property (nonatomic,strong)UIColor * yjdlShAppPrivacyWebNavigationTintColor;
/**UINavigationBarTintColor*/
@property (nonatomic,strong)UIColor * yjdlShAppPrivacyWebNavigationBarTintColor;
/**UINavigationBackgroundImage*/
@property (nonatomic,strong)UIImage * yjdlShAppPrivacyWebNavigationBackgroundImage;
/**UINavigationBarMetrics*/
@property (nonatomic,strong)NSNumber * yjdlShAppPrivacyWebNavigationBarMetrics;
/**UINavigationShadowImage*/
@property (nonatomic,strong)UIImage * yjdlShAppPrivacyWebNavigationShadowImage;
/**UINavigationBarStyle*/
@property (nonatomic,strong)NSNumber * yjdlShAppPrivacyWebNavigationBarStyle;
/*SLOGAN
注： 运营商品牌标签("中国**提供认证服务")，不得隐藏
**/
/**slogan文字字体*/
@property (nonatomic,strong) UIFont * yjdlShSloganTextFont;
/**slogan文字颜色*/
@property (nonatomic,strong) UIColor * yjdlShSloganTextColor;
/**slogan文字对齐方式 NSTextAlignment eg.@(NSTextAlignmentCenter)*/
@property (nonatomic,strong) NSNumber * yjdlShSlogaTextAlignment;
/*验证SLOGAN
注： 供应商品牌标签("验证提供认技术支持")
**/
/**slogan文字字体*/
@property (nonatomic,strong) UIFont * yjdlShSupplierSloganTextFont;
```

```
/**slogan文字颜色*/
@property (nonatomic,strong) UIColor * yjdlShSupplierSloganTextColor;
/**slogan文字对齐方式 NSTextAlignment eg.@(NSTextAlignmentCenter)*/
@property (nonatomic,strong) NSNumber * yjdlShSupplierSloganTextAlignment;
/**slogan默认不隐藏 eg.@(NO)*/
@property (nonatomic,strong) NSNumber * yjdlShSupplierSloganHidden;
/*CheckBox
*协议勾选框，默认选中且在协议前显示
*可在sdk_oauth.bundle中替换checkBox_unSelected、checkBox_selected图片
*也可以通过属性设置选中和未选择图片
**/
/**协议勾选框（默认显示,放置在协议之前）BOOL eg.@(YES)*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxHidden;
/**协议勾选框默认值（默认选中）BOOL eg.@(YES)*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxValue;
/**协议勾选框 尺寸 NSValue->CGSize eg.[NSValue valueWithCGSize:CGSizeMake(25, 25)]*/
@property (nonatomic,strong) NSValue *yjdlShCheckBoxSize;
/**协议勾选框 UIButton.image图片缩进 UIEdgeInset eg.[NSValue valueWithUIEdgeInsets:UIEdgeInsetsMake(2, 2, 2, 2)]*/
@property (nonatomic,strong) NSValue *yjdlShCheckBoxImageEdgeInsets;
/**协议勾选框 设置CheckBox顶部与隐私协议控件顶部对齐 YES或大于0生效 eg.@(YES)*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentToAppPrivacyTop;
/**协议勾选框 设置CheckBox对齐后的偏移量,相对于对齐后的中心距离在当前垂直方向上的偏移*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentOffset;
/**协议勾选框 设置CheckBox顶部与隐私协议控件竖向中心对齐 YES或大于0生效 eg.@(YES)*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentToAppPrivacyCenterY;
/**协议勾选框 非选中状态图片*/
@property (nonatomic,strong) UIImage *yjdlShCheckBoxUncheckedImage;
/**协议勾选框 选中状态图片*/
@property (nonatomic,strong) UIImage *yjdlShCheckBoxCheckedImage;
/**授权页自定义 "请勾选协议"提示框
- containerView为loading的全屏蒙版view
- 请自行在containerView添加自定义提示
*/
@property (nonatomic,copy)void(^checkBoxTipView)(UIView * containerView);
/**checkBox 未勾选时 提示文本，默认："请勾选协议"*/
@property (nonatomic,copy) NSString *yjdlShCheckBoxTipMsg;
/**使用sdk内部“一键登录”按钮点击时的吐丝提示("请勾选协议")
* NO:默认使用sdk内部吐丝 YES:禁止使用
*/
@property (nonatomic,strong) NSNumber *yjdlShCheckBoxTipDisable;
```

```
/*Loading*/
/**Loading 大小 CGSize eg.[NSValue valueWithCGSize:CGSizeMake(50, 50)]*/
@property (nonatomic,strong) NSValue *yjdlShLoadingSize;
/**Loading 圆角 float eg.@(5) */
@property (nonatomic,strong) NSNumber *yjdlShLoadingCornerRadius;
/**Loading 背景色 UIColor eg.[UIColor colorWithRed:0.8 green:0.5 blue:0.8 alpha:0.8]; */
@property (nonatomic,strong) UIColor *yjdlShLoadingBackgroundColor;
/**UIActivityIndicatorViewStyle eg.@(UIActivityIndicatorViewStyleWhiteLarge)*/
@property (nonatomic,strong) NSNumber *yjdlShLoadingIndicatorStyle;
/**Loading Indicator渲染色 UIColor eg.[UIColor greenColor]; */
@property (nonatomic,strong) UIColor *yjdlShLoadingTintColor;
/**授权页自定义Loading
- containerView为loading的全屏蒙版view
- 请自行在containerView添加自定义loading
- 设置block后，上述loading属性将无效
*/
@property (nonatomic,copy)void(\^loadingView)(UIView * containerView);
//添加自定义控件
/**可设置背景色及添加控件*/
@property (nonatomic,copy)void(\^customAreaView)(UIView * customAreaView);
/**设置隐私协议弹窗*/
@property (nonatomic,copy)void(\^customPrivacyAlertView)(UIViewController * authPageVC);
/**横竖屏*/
/*是否支持自动旋转 BOOL*/
@property (nonatomic,strong) NSNumber * shouldAutorotate;
/*支持方向 UIInterfaceOrientationMask
- 如果设置只支持竖屏，只需设置YjdlShOrientationLayOutPortrait竖屏布局对象
- 如果设置只支持横屏，只需设置YjdlShOrientationLayOutLandscape横屏布局对象
- 横竖屏均支持，需同时设置YjdlShOrientationLayOutPortrait和YjdlShOrientationLayOutLandscape
*/
@property (nonatomic,strong) NSNumber * supportedInterfaceOrientations;
/*默认方向 UIInterfaceOrientation*/
@property (nonatomic,strong) NSNumber * preferredInterfaceOrientationForPresentation;
/**以窗口方式显示授权页
*/
/**以窗口方式显示 BOOL, default is NO */
@property (nonatomic,strong) NSNumber * yjdlShAuthTypeUseWindow;
/**窗口圆角 float*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowCornerRadius;
/**yjdlShAuthWindowModalTransitionStyle系统自带的弹出方式 仅支持以下三种
```

```
UIModalTransitionStyleCoverVertical 底部弹出
UIModalTransitionStyleCrossDissolve 淡入
UIModalTransitionStyleFlipHorizontal 翻转显示
*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowModalTransitionStyle;
/* UIModalPresentationStyle
* 若使用窗口模式，请设置为UIModalPresentationOverFullScreen 或不设置
* iOS13强制全屏，请设置为UIModalPresentationFullScreen
* UIModalPresentationAutomatic API_AVAILABLE(ios(13.0)) = -2
* 默认UIModalPresentationFullScreen
* eg. @(UIModalPresentationOverFullScreen)
*/
/*授权页 ModalPresentationStyle*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowModalPresentationStyle;
/*协议页 ModalPresentationStyle （授权页使用窗口模式时，协议页强制使用模态弹出）*/
@property (nonatomic,strong) NSNumber * yjdlShAppPrivacyWebModalPresentationStyle;
/* UIUserInterfaceStyle
* UIUserInterfaceStyleUnspecified - 不指定样式，跟随系统设置进行展示
* UIUserInterfaceStyleLight - 明亮
* UIUserInterfaceStyleDark, - 暗黑 仅对iOS13+系统有效
*/
/*授权页 UIUserInterfaceStyle,默认:UIUserInterfaceStyleLight,eg. @(UIUserInterfaceStyleLight)*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowOverrideUserInterfaceStyle;
/**
* 授权页面present弹出时animate动画设置，默认带动画，eg. @(YES)
*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowPresentingAnimate;
/**
* sdk自带返回键：授权页面dismiss时animate动画设置，默认带动画，eg. @(YES)
*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowDismissAnimate;
/**弹窗的MaskLayer，用于自定义窗口形状*/
@property (nonatomic,strong) CALayer * yjdlShAuthWindowMaskLayer;
//竖屏布局配置对象 -->创建一个布局对象，设置好控件约束属性值，再设置到此属性中
/**竖屏：UIInterfaceOrientationPortrait|UIInterfaceOrientationPortraitUpsideDown
*eg. YjdlSUIConfigure* baseUIConfigure = [YjdlSUIConfigurenew];
* YjdlShOrientationLayOut * YjdlShOrientationLayOutPortrait = [YjdlShOrientationLayOut new];
* YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneCenterY = @(0);
* YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneLeft = @(50*screenScale);
* ...
```

```
* baseUIConfigure.YjdlShOrientationLayOutPortrait = YjdlShOrientationLayOutPortrait;
*/
@property (nonatomic,strong) YjdlShOrientationLayOut * YjdlShOrientationLayOutPortrait;
//横屏布局配置对象 -->创建一个布局对象，设置好控件约束属性值，再设置到此属性中
/**横屏：UIInterfaceOrientationLandscapeLeft|UIInterfaceOrientationLandscapeRight
*eg. YjdlSUIConfigure* baseUIConfigure = [YjdlSUIConfigure new];
* YjdlShOrientationLayOut * YjdlShOrientationLayOutLandscape = [YjdlShOrientationLayOut new];
* YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneCenterY = @(0);
* YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneLeft = @(50*screenScale);
* ...
* baseUIConfigure.YjdlShOrientationLayOutLandscape = YjdlShOrientationLayOutLandscape;
*/
@property (nonatomic,strong) YjdlShOrientationLayOut * YjdlShOrientationLayOutLandscape;
/**默认界面配置*/
+ (YjdlSUIConfigure *)yjdlShDefaultUIConfigure;
@end
/**横竖屏布局配置对象
配置页面布局相关属性
*/
@interface YjdlShOrientationLayOut : NSObject
/**LOGO图片*/
// 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoLeft;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLogoCenterY;
/**手机号显示控件*/
//layout 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneLeft;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutPhoneCenterY;
```

```
/*一键登录按钮 控件
注： 一键登录授权按钮 不得隐藏
**/
//layout 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnLeft;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutLoginBtnCenterY;
/*隐私条款Privacy
注： 运营商隐私条款 不得隐藏， 用户条款不限制
**/
//layout 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyLeft;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutAppPrivacyCenterY;
/*Slogan 运营商品牌标签："认证服务由中国移动/联通/电信提供" label
注： 运营商品牌标签，不得隐藏
**/
//layout 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganLeft;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutSloganCenterY;
/*验证Slogan 供应商品牌标签："验证提供技术支持" label
**/
//layout 约束均相对vc.view
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganLeft;
```

```
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganTop;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganRight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganBottom;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganWidth;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganHeight;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganCenterX;
@property (nonatomic,strong)NSNumber * yjdlShLayoutVerifySloganCenterY;
/**窗口模式*/
/**窗口中心：CGPoint X Y*/
@property (nonatomic,strong) NSValue * yjdlShAuthWindowOrientationCenter;
/**窗口左上角：frame.origin：CGPoint X Y*/
@property (nonatomic,strong) NSValue * yjdlShAuthWindowOrientationOrigin;
/**窗口大小：宽 float */
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowOrientationWidth;
/**窗口大小：高 float */
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowOrientationHeight;
```

3.横竖屏设置

相关设置属性

```
/**横竖屏*/
/*是否支持自动旋转 BOOL*/
@property (nonatomic,strong) NSNumber * shouldAutorotate;
/*支持方向 UIInterfaceOrientationMask
- 如果设置只支持竖屏，只需设置YjdlShOrientationLayOutPortrait竖屏布局对象
- 如果设置只支持横屏，只需设置YjdlShOrientationLayOutLandscape横屏布局对象
- 横竖屏均支持，需同时设置YjdlShOrientationLayOutPortrait和YjdlShOrientationLayOutLandscape
*/
@property (nonatomic,strong) NSNumber * supportedInterfaceOrientations;
/*默认方向 UIInterfaceOrientation*/
@property (nonatomic,strong) NSNumber * preferredInterfaceOrientationForPresentation;
```

使用示例代码-请下载demo查看详细配置

```
-(YjdlSUIConfigure*)configureMake5{
//黑
UIColor *color1 = [UIColor colorWithRed:30/255.0 green:30/255.0 blue:30/255.0 alpha:1.0];
//灰
UIColor *color2 = [UIColor colorWithRed:100/255.0 green:100/255.0 blue:100/255.0 alpha:1.0];
//天蓝
UIColor *color3 = [UIColor colorWithRed:60/255.0 green:160/255.0 blue:247/255.0 alpha:1.0];
YjdlSUIConfigure*configure = [[YjdlSUIConfigurealloc] init];
configure.viewController = self;
configure.shouldAutorotate = @(YES);
```

```
//导航设置
configure.yjdlShNavigationBarHidden = @(YES);
//logo
configure.yjdlShLogoHiden = @(YES);
//掩码
configure.yjdlShPhoneNumberFont = [UIFont systemFontOfSize:22];
configure.yjdlShPhoneNumberColor = color1;
configure.yjdlShPhoneNumberTextAlignment = @(NSTextAlignmentCenter);
//slog
configure.yjdlShSloganTextFont = [UIFont systemFontOfSize:14];
configure.yjdlShSloganTextColor = color2;
configure.yjdlShSlogaTextAlignment = @(NSTextAlignmentCenter);
//登录按钮
configure.yjdlShLoginBtnText = @"本机号一键登录";
configure.yjdlShLoginBtnTextFont = [UIFont systemFontOfSize:18 weight:0.2];
configure.yjdlShLoginBtnTextColor = [UIColor whiteColor];
configure.yjdlShLoginBtnBgColor = color3;
configure.yjdlShLoginBtnCornerRadius = @(5);
//协议
configure.yjdlShCheckBoxCheckedImage = [UIImage imageNamed:@"checkbox-multiple-ma"];
configure.yjdlShCheckBoxUncheckedImage = [UIImage imageNamed:@"checkbox-multiple-bl"];
configure.yjdlShCheckBoxSize = [NSValue valueWithCGSize:CGSizeMake(25, 25)];
configure.yjdlShCheckBoxImageEdgeInsets = [NSValue valueWithUIEdgeInsets:UIEdgeInsetsMake(0, 0, 5, 5)];
configure.yjdlShCheckBoxValue = @(NO);
configure.yjdlShAppPrivacyNormalDesTextFirst = @"我已阅读并同意";
configure.yjdlShAppPrivacyFirst = @[@"《用户协议》",[NSURL URLWithString:@"https://baidu.com"]];
configure.yjdlShAppPrivacyNormalDesTextSecond = @"、";
configure.yjdlShAppPrivacySecond = @[@"《隐私政策》",[NSURL URLWithString:@"https://baidu.com"]];
configure.yjdlShAppPrivacyNormalDesTextThird = @"和";
configure.yjdlShAppPrivacyTextFont = [UIFont systemFontOfSize:14];
configure.yjdlShAppPrivacyTextAlignment = @(NSTextAlignmentLeft);
configure.yjdlShAppPrivacyLineSpacing = @(3.0);
configure.yjdlShOperatorPrivacyAtLast = @(YES);
configure.yjdlShAppPrivacyPunctuationMarks = @(YES);
configure.yjdlShAppPrivacyColor = @[color2,color3];
configure.yjdlShAuthTypeUseWindow = @(YES);
CGFloat width = MIN([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);
CGFloat height = MAX([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);
CGFloat scale = width/375.0;//以iphone6 屏幕为基准
UIInterfaceOrientation orientation = [UIApplication sharedApplication].statusBarOrientation;
```

```
YjdlShOrientationLayOut *YjdlShOrientationLayOutPortrait = [[YjdlShOrientationLayOut alloc] init];

configure.YjdlShOrientationLayOutPortrait = YjdlShOrientationLayOutPortrait;

CGFloat top = height/2.0 - 340*scale/2.0;

//手机掩码

YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneTop = @(top + 70*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneLeft = @(50*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneCenterX = @(0);

YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneHeight = @(30*scale);

top = YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneTop.floatValue +
YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneHeight.floatValue;

//slog

YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganTop = @(top + 20*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganLeft = @(50*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganCenterX = @(0);

YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganHeight = @(25*scale);

top = YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganTop.floatValue +
YjdlShOrientationLayOutPortrait.yjdlShLayoutSloganHeight.floatValue;

//登录按钮

YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnTop = @(top + 25*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnLeft = @(50*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnRight = @(-50*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnHeight = @(45*scale);

top = YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnTop.floatValue +
YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnHeight.floatValue;

//协议

YjdlShOrientationLayOutPortrait.yjdlShLayoutAppPrivacyTop = @(top + 70*scale);

YjdlShOrientationLayOutPortrait.yjdlShLayoutAppPrivacyLeft = @(50*scale + 30);

YjdlShOrientationLayOutPortrait.yjdlShLayoutAppPrivacyRight = @(-50*scale);

top = YjdlShOrientationLayOutPortrait.yjdlShLayoutAppPrivacyTop.floatValue +
YjdlShOrientationLayOutPortrait.yjdlShLayoutAppPrivacyHeight.floatValue;

__weak typeof(self) weakSelf = self;

//横屏

CGFloat heightLandscape = MIN([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);

CGFloat widthLandscape = MAX([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);

CGFloat topLandscape = heightLandscape/2.0 - 300*scale/2.0;

YjdlShOrientationLayOut *YjdlShOrientationLayOutLandscape = [[YjdlShOrientationLayOut alloc] init];

configure.YjdlShOrientationLayOutLandscape = YjdlShOrientationLayOutLandscape;

//手机掩码

YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneTop = @(topLandscape + 60*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneCenterX = @(0);

YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneWidth = @(heightLandscape - 100*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneHeight = @(30*scale);
```

```
topLandscape = YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneTop.floatValue +
YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneHeight.floatValue;

//slog

YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganTop = @(topLandscape + 10*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganWidth = @(heightLandscape - 100*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganCenterX = @(0);

YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganHeight = @(20*scale);

topLandscape = YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganTop.floatValue +
YjdlShOrientationLayOutLandscape.yjdlShLayoutSloganHeight.floatValue;

//登录按钮

YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnTop = @(topLandscape + 15*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnWidth = @(heightLandscape - 100*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnCenterX = @(0);

YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnHeight = @(45*scale);

topLandscape = YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnTop.floatValue +
YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnHeight.floatValue;

//协议

YjdlShOrientationLayOutLandscape.yjdlShLayoutAppPrivacyTop = @(topLandscape + 60*scale);

YjdlShOrientationLayOutLandscape.yjdlShLayoutAppPrivacyWidth = @(heightLandscape - 100*scale-30);

YjdlShOrientationLayOutLandscape.yjdlShLayoutAppPrivacyCenterX = @(15);

topLandscape = YjdlShOrientationLayOutLandscape.yjdlShLayoutAppPrivacyTop.floatValue +
YjdlShOrientationLayOutLandscape.yjdlShLayoutAppPrivacyHeight.floatValue;

__block UIView *conerViewBlock;

__block UIButton *yjdlShoseBlock;

__block UILabel *welcomLabelBlock;

__block UIButton *otherButtonBlock;

configure.customAreaView = \^(UIView * _Nonnull customAreaView) {

customAreaView.backgroundColor = [[UIColor blackColor] colorWithAlphaComponent:0.3];

void (\^ rientationPortraitBlcok) (UIInterfaceOrientation orientation) = \^(UIInterfaceOrientation orientation){

UIView *conerView = [[UIView alloc] init];

conerView.backgroundColor = [UIColor whiteColor];

conerView.layer.cornerRadius = 10.0;

[customAreaView addSubview:conerView];

conerViewBlock = conerView;

//关闭按钮

UIButton *yjdlShose = [UIButton buttonWithType:UIButtonTypeCustom];

[yjdlShose setBackgroundImage:[UIImage imageNamed:@"back"] forState:UIControlStateNormal];

[yjdlShose addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];

[conerView addSubview:yjdlShose];

yjdlShoseBlock = yjdlShose;

UILabel *welcomLabel = [[UILabel alloc] init];

welcomLabel.text = @"欢迎使用一键登录";
```

```
welcomLabel.font = [UIFont systemFontOfSize:15];
welcomLabel.textColor = color1;
welcomLabel.textAlignment = NSTextAlignmentCenter;
[customAreaView addSubview:welcomLabel];
welcomLabelBlock = welcomLabel;
UIButton *otherButton = [UIButton buttonWithType:UIButtonTypeCustom];
[otherButton setTitle:@"其他方式登录" forState:UIControlStateNormal];
[otherButton setTitleColor:color1 forState:UIControlStateNormal];
[otherButton.titleLabel setFont:[UIFont systemFontOfSize:15.0]];
[otherButton addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];
[customAreaView addSubview:otherButton];
otherButtonBlock = otherButton;
//竖屏
if (orientation == UIInterfaceOrientationPortrait || orientation == UIInterfaceOrientationPortraitUpsideDown) {
[conerView mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_equalTo(YjdlShOrientationLayOutPortrait.yjdlShLayoutPhoneTop.floatValue - 70*scale);
make.left.mas_equalTo(30*scale);
make.centerX.mas_equalTo(0);
make.height.mas_equalTo(340*scale + 35*scale);
}];
[yjdlShose mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_offset(10*scale);
make.right.mas_offset(-10*scale);
make.width.height.mas_equalTo(25*scale);
}];
[welcomLabel mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerY.mas_equalTo(conerView.mas_top).offset(40*scale);
make.left.mas_equalTo(50*scale);
make.centerX.mas_equalTo(0);
make.height.mas_offset(25*scale);
}];
//其他方式登录
[otherButton mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerX.mas_equalTo(0);
make.centerY.equalTo(customAreaView.mas_top).offset(YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnTop.floatValue
+ YjdlShOrientationLayOutPortrait.yjdlShLayoutLoginBtnHeight.floatValue + 35*scale);
}];
}else{
//横屏
[conerView mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_equalTo(YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneTop.floatValue - 60*scale);
```

```
make.width.mas_equalTo(heightLandscape - 60*scale);
make.centerX.mas_equalTo(0);
make.bottom.mas_offset(-YjdlShOrientationLayOutLandscape.yjdlShLayoutPhoneTop.floatValue + 60*scale+10*scale);
}];
[yjdlShose mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_offset(10*scale);
make.right.mas_offset(-10*scale);
make.width.height.mas_equalTo(25*scale);
}];
[welcomLabel mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerY.mas_equalTo(conerView.mas_top).offset(40*scale);
make.left.mas_equalTo(50*scale);
make.centerX.mas_equalTo(0);
make.height.mas_offset(25*scale);
}];
//其他方式登录
[otherButton mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerX.mas_equalTo(0);
make.centerY.equalTo(customAreaView.mas_top).offset(YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnTop.floatValue
+ YjdlShOrientationLayOutLandscape.yjdlShLayoutLoginBtnHeight.floatValue + 35*scale);
}];
}
};
rientationPortraitBlcok(orientation);
[[NSNotificationCenter defaultCenter] addObserverForName:UIApplicationDidChangeStatusBarOrientationNotification
object:nil queue:NSOperationQueue.mainQueue usingBlock:\^(NSNotification * _Nonnull note) {
UIInterfaceOrientation orientationIn = [UIApplication sharedApplication].statusBarOrientation;
//删除以前约束
[conerViewBlock removeFromSuperview];
[yjdlShoseBlock removeFromSuperview];
[welcomLabelBlock removeFromSuperview];
[otherButtonBlock removeFromSuperview];
rientationPortraitBlcok(orientationIn);
}];
};
return configure;
}
```

4.弹窗模式设置

4.1 相关设置属性

```
/**以窗口方式显示授权页
*/
```

```
/**以窗口方式显示 BOOL, default is NO */
@property (nonatomic,strong) NSNumber * yjdlShAuthTypeUseWindow;
/**窗口圆角 float*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowCornerRadius;
/**yjdlShAuthWindowModalTransitionStyle系统自带的弹出方式 仅支持以下三种
UIModalTransitionStyleCoverVertical 底部弹出
UIModalTransitionStyleCrossDissolve 淡入
UIModalTransitionStyleFlipHorizontal 翻转显示
*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowModalTransitionStyle;
/* UIModalPresentationStyle
* 若使用窗口模式，请设置为UIModalPresentationOverFullScreen 或不设置
* iOS13强制全屏，请设置为UIModalPresentationFullScreen
* UIModalPresentationAutomatic API_AVAILABLE(ios(13.0)) = -2
* 默认UIModalPresentationFullScreen
* eg. @(UIModalPresentationOverFullScreen)
*/
/*授权页 ModalPresentationStyle*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowModalPresentationStyle;
/*协议页 ModalPresentationStyle （授权页使用窗口模式时，协议页强制使用模态弹出）*/
@property (nonatomic,strong) NSNumber * yjdlShAppPrivacyWebModalPresentationStyle;
/**弹窗的MaskLayer，用于自定义窗口形状*/
@property (nonatomic,strong) CALayer * yjdlShAuthWindowMaskLayer;
```

4.2 布局示例代码

详细请下载demo查看配置

```
-(YjdlSUIConfigure*)configureMake3{
//黑
UIColor *color1 = [UIColor colorWithRed:30/255.0 green:30/255.0 blue:30/255.0 alpha:1.0];
//灰
UIColor *color2 = [UIColor colorWithRed:100/255.0 green:100/255.0 blue:100/255.0 alpha:1.0];
//天蓝
UIColor *color3 = [UIColor colorWithRed:60/255.0 green:160/255.0 blue:247/255.0 alpha:1.0];
YjdlSUIConfigure*configure = [[YjdlSUIConfigurealloc] init];
configure.viewController = self;
configure.shouldAutorotate = @(NO);;
//导航设置
configure.yjdlShNavigationBarHidden = @(YES);
//logo
configure.yjdlShLogoHiden = @(YES);
//掩码
configure.yjdlShPhoneNumberFont = [UIFont systemFontOfSize:22];
```

```
configure.yjdlShPhoneNumberColor = color1;
configure.yjdlShPhoneNumberTextAlignment = @(NSTextAlignmentCenter);
//slog
configure.yjdlShSloganTextFont = [UIFont systemFontOfSize:14];
configure.yjdlShSloganTextColor = color2;
configure.yjdlShSlogaTextAlignment = @(NSTextAlignmentCenter);
//登录按钮
configure.yjdlShLoginBtnText = @"本机号一键登录";
configure.yjdlShLoginBtnTextFont = [UIFont systemFontOfSize:18 weight:0.2];
configure.yjdlShLoginBtnTextColor = [UIColor whiteColor];
configure.yjdlShLoginBtnBgColor = color3;
configure.yjdlShLoginBtnCornerRadius = @(5);
//协议
configure.yjdlShCheckBoxCheckedImage = [UIImage imageNamed:@"checkbox-multiple-ma"];
configure.yjdlShCheckBoxUncheckedImage = [UIImage imageNamed:@"checkbox-multiple-bl"];
configure.yjdlShCheckBoxSize = [NSValue valueWithCGSize:CGSizeMake(25, 25)];
configure.yjdlShCheckBoxImageEdgeInsets = [NSValue valueWithUIEdgeInsets:UIEdgeInsetsMake(0, 0, 5, 5)];
configure.yjdlShCheckBoxValue = @(NO);
configure.yjdlShAppPrivacyNormalDesTextFirst = @"我已阅读并同意";
configure.yjdlShAppPrivacyFirst = @[@"《用户协议》",[NSURL URLWithString:@"https://baidu.com"]];
configure.yjdlShAppPrivacyNormalDesTextSecond = @"、";
configure.yjdlShAppPrivacySecond = @[@"《隐私政策》",[NSURL URLWithString:@"https://baidu.com"]];
configure.yjdlShAppPrivacyNormalDesTextThird = @"和";
configure.yjdlShAppPrivacyTextFont = [UIFont systemFontOfSize:14];
configure.yjdlShAppPrivacyTextAlignment = @(NSTextAlignmentLeft);
configure.yjdlShAppPrivacyLineSpacing = @(3.0);
configure.yjdlShOperatorPrivacyAtLast = @(YES);
configure.yjdlShAppPrivacyPunctuationMarks = @(YES);
configure.yjdlShAppPrivacyColor = @[color2,color3];
configure.yjdlShAuthTypeUseWindow = @(YES);
CGFloat width = MIN([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);
CGFloat height = MAX([UIScreen mainScreen].bounds.size.width, [UIScreen mainScreen].bounds.size.height);
CGFloat scale = width/375.0;//以iphone6 屏幕为基准
YjdlShOrientationLayOut *YjdlShOrientationLayOut = [[YjdlShOrientationLayOut alloc] init];
configure.YjdlShOrientationLayOutPortrait = YjdlShOrientationLayOut;
CGFloat top = height/2.0 - 340*scale/2.0;
//手机掩码
YjdlShOrientationLayOut.yjdlShLayoutPhoneTop = @(top + 70*scale);
YjdlShOrientationLayOut.yjdlShLayoutPhoneLeft = @(50*scale);
YjdlShOrientationLayOut.yjdlShLayoutPhoneCenterX = @(0);
```

```
YjdlShOrientationLayOut.yjdlShLayoutPhoneHeight = @(30*scale);

top = YjdlShOrientationLayOut.yjdlShLayoutPhoneTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutPhoneHeight.floatValue;

//slog

YjdlShOrientationLayOut.yjdlShLayoutSloganTop = @(top + 20*scale);

YjdlShOrientationLayOut.yjdlShLayoutSloganLeft = @(50*scale);

YjdlShOrientationLayOut.yjdlShLayoutSloganCenterX = @(0);

YjdlShOrientationLayOut.yjdlShLayoutSloganHeight = @(25*scale);

top = YjdlShOrientationLayOut.yjdlShLayoutSloganTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutSloganHeight.floatValue;

//登录按钮

YjdlShOrientationLayOut.yjdlShLayoutLoginBtnTop = @(top + 25*scale);

YjdlShOrientationLayOut.yjdlShLayoutLoginBtnLeft = @(50*scale);

YjdlShOrientationLayOut.yjdlShLayoutLoginBtnRight = @(-50*scale);

YjdlShOrientationLayOut.yjdlShLayoutLoginBtnHeight = @(45*scale);

top = YjdlShOrientationLayOut.yjdlShLayoutLoginBtnTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutLoginBtnHeight.floatValue;

//协议

YjdlShOrientationLayOut.yjdlShLayoutAppPrivacyTop = @(top + 70*scale);

YjdlShOrientationLayOut.yjdlShLayoutAppPrivacyLeft = @(50*scale + 30);

YjdlShOrientationLayOut.yjdlShLayoutAppPrivacyRight = @(-50*scale);

top = YjdlShOrientationLayOut.yjdlShLayoutAppPrivacyTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutAppPrivacyHeight.floatValue;

__weak typeof(self) weakSelf = self;

configure.customAreaView = \^(UIView * _Nonnull customAreaView) {

customAreaView.backgroundColor = [[UIColor blackColor] colorWithAlphaComponent:0.3];

UIView *conerView = [[UIView alloc] init];

conerView.backgroundColor = [UIColor whiteColor];

conerView.layer.cornerRadius = 10.0;

[customAreaView addSubview:conerView];

[conerView mas_makeConstraints:\^(MASConstraintMaker *make) {

make.top.mas_equalTo(YjdlShOrientationLayOut.yjdlShLayoutPhoneTop.floatValue - 70*scale);

make.left.mas_equalTo(30*scale);

make.centerX.mas_equalTo(0);

make.height.mas_equalTo(340*scale + 35*scale);

}];

//关闭按钮

UIButton *yjdlShose = [UIButton buttonWithType:UIButtonTypeCustom];

[yjdlShose setBackgroundImage:[UIImage imageNamed:@"back"] forState:UIControlStateNormal];

[yjdlShose addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];

[conerView addSubview:yjdlShose];

[yjdlShose mas_makeConstraints:\^(MASConstraintMaker *make) {
```

```
make.top.mas_offset(10*scale);
make.right.mas_offset(-10*scale);
make.width.height.mas_equalTo(25*scale);
}];
UILabel *welcomLabel = [[UILabel alloc] init];
welcomLabel.text = @"欢迎使用一键登录";
welcomLabel.font = [UIFont systemFontOfSize:15];
welcomLabel.textColor = color1;
welcomLabel.textAlignment = NSTextAlignmentCenter;
[customAreaView addSubview:welcomLabel];
[welcomLabel mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerY.mas_equalTo(conerView.mas_top).offset(40*scale);
make.left.mas_equalTo(50*scale);
make.centerX.mas_equalTo(0);
make.height.mas_offset(25*scale);
}];
//其他方式登录
UIButton *otherButton = [UIButton buttonWithType:UIButtonTypeCustom];
[otherButton setTitle:@"其他方式登录" forState:UIControlStateNormal];
[otherButton setTitleColor:color1 forState:UIControlStateNormal];
[otherButton.titleLabel setFont:[UIFont systemFontOfSize:15.0]];
[otherButton addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];
[customAreaView addSubview:otherButton];
[otherButton mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerX.mas_equalTo(0);
make.centerY.equalTo(customAreaView.mas_top).offset(YjdlShOrientationLayOut.yjdlShLayoutLoginBtnTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutLoginBtnHeight.floatValue + 35*scale);
}];
};
return configure;
}
```

5.添加自定义控件示例

ObjC：

```
// 快捷登录
- (void)quickLoginBtnClick:(UIButton *)sender {
...
configure.customAreaView = \^(UIView * _Nonnull customAreaView) {
customAreaView.backgroundColor = [[UIColor blackColor] colorWithAlphaComponent:0.3];
UIView *conerView = [[UIView alloc] init];
conerView.backgroundColor = [UIColor whiteColor];
conerView.layer.cornerRadius = 10.0;
```

```
[customAreaView addSubview:conerView];
[conerView mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_equalTo(YjdlShOrientationLayOut.yjdlShLayoutPhoneTop.floatValue - 70*scale);
make.left.mas_equalTo(30*scale);
make.centerX.mas_equalTo(0);
make.height.mas_equalTo(340*scale + 35*scale);
}];
//关闭按钮
UIButton *yjdlShose = [UIButton buttonWithType:UIButtonTypeCustom];
[yjdlShose setBackgroundImage:[UIImage imageNamed:@"back"] forState:UIControlStateNormal];
[yjdlShose addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];
[conerView addSubview:yjdlShose];
[yjdlShose mas_makeConstraints:\^(MASConstraintMaker *make) {
make.top.mas_offset(10*scale);
make.right.mas_offset(-10*scale);
make.width.height.mas_equalTo(25*scale);
}];
UILabel *welcomLabel = [[UILabel alloc] init];
welcomLabel.text = @"欢迎使用一键登录";
welcomLabel.font = [UIFont systemFontOfSize:15];
welcomLabel.textColor = color1;
welcomLabel.textAlignment = NSTextAlignmentCenter;
[customAreaView addSubview:welcomLabel];
[welcomLabel mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerY.mas_equalTo(conerView.mas_top).offset(40*scale);
make.left.mas_equalTo(50*scale);
make.centerX.mas_equalTo(0);
make.height.mas_offset(25*scale);
}];
//其他方式登录
UIButton *otherButton = [UIButton buttonWithType:UIButtonTypeCustom];
[otherButton setTitle:@"其他方式登录" forState:UIControlStateNormal];
[otherButton setTitleColor:color1 forState:UIControlStateNormal];
[otherButton.titleLabel setFont:[UIFont systemFontOfSize:15.0]];
[otherButton addTarget:weakSelf action:@selector(dismis) forControlEvents:UIControlEventTouchUpInside];
[customAreaView addSubview:otherButton];
[otherButton mas_makeConstraints:\^(MASConstraintMaker *make) {
make.centerX.mas_equalTo(0);
make.centerY.equalTo(customAreaView.mas_top).offset(YjdlShOrientationLayOut.yjdlShLayoutLoginBtnTop.floatValue +
YjdlShOrientationLayOut.yjdlShLayoutLoginBtnHeight.floatValue + 35*scale);
}];
```

```
};
...
}
//授权页 点击自定义控件绑定的方法
-(void)otherLoginWayBtnCliced:(UIButton *)sender{
//关闭页面
[YjdlShSDKManager finishAuthControllerCompletion:\^{
//如需关闭后present/push新页面，建议在completion回调中执行
}];
}
```

6.添加自定义loading

```
// 自定义loading--将loadingView添加到containerView上进行显示
baseUIConfigure.loadingView = \^(UIView * _Nonnull containerView) {
UIImage *image = [YYImage imageNamed:@"KungFuPanda.gif"];
UIImageView *imageBackground = [[YYAnimatedImageView alloc] initWithImage:image];
[containerView addSubview:imageBackground];
[imageBackground mas_makeConstraints:\^(MASConstraintMaker *make) {
make.size.mas_equalTo(CGSizeMake(80, 80));
make.center.mas_equalTo(0);
}];
imageBackground.layer.cornerRadius = 40;
imageBackground.layer.masksToBounds = YES;
};
```

7.自定义设置隐私协议弹窗

```
...
baseUIConfigure.customPrivacyAlertView = \^(UIViewController * _Nonnull authPageVC) {
IAlertController * privacyAlert = [UIAlertController alertControllerWithTitle:@"用户注册及使用APP隐私协议"
message:@"在此特别提醒您（用户）在注册成为用户之前，请认真阅读本《用户协议》（以下简称“协议”），确保您充分理解本协议中各条款。请您审慎阅读并选择接受或不接受本协议。您的注册、登录、使用等行为将视为对本协议的接受，并同意接受本协议各项条款的约束。本协议约定xxxx用本服务的个人。本协议可由喵星随时更新，更新后的协议条款一旦公布即代替原来的协议条款，恕不再另行通知，用户可在本APP中查阅最新版协议条款。在修改协议条款后，如果用户不接受修改后的条款，请立即停止使用喵小瞳提供的服务，用户继续使用服务将被视为接受修改后的协议。 " preferredStyle:(UIAlertControllerStyleAlert)];
[privacyAlert addAction:[UIAlertAction actionWithTitle:@"拒绝" style:(UIAlertActionStyleCancel)
handler:\^(UIAlertAction * _Nonnull action) {
@strongify(self)
[self hideAuthPageMaskViewWhenUseWindow];
[YjdlShSDKManager finishAuthControllerCompletion:nil];
}]];
[privacyAlert addAction:[UIAlertAction actionWithTitle:@"同意" style:(UIAlertActionStyleDefault)
handler:\^(UIAlertAction * _Nonnull action) {
//通知sdk勾选协议
[YjdlShSDKManager setCheckBoxValue:YES];
}]];
[authPageVC presentViewController:privacyAlert animated:YES completion:nil];
```

```
};
...
```

8.适配iOS13

弹出风格

```
/* UIModalPresentationStyle
* 若使用窗口模式，请设置为UIModalPresentationOverFullScreen 或不设置
* iOS13强制全屏，请设置为UIModalPresentationFullScreen
* UIModalPresentationAutomatic API_AVAILABLE(ios(13.0)) = -2
* eg. @(UIModalPresentationOverFullScreen)
*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowModalPresentationStyle;
```

状态栏

```
/*状态栏样式
*Info.plist: View controller-based status bar appearance = YES
*
*UIStatusBarStyleDefault：状态栏显示 黑
*UIStatusBarStyleLightContent：状态栏显示 白
*UIStatusBarStyleDarkContent：状态栏显示 黑 API_AVAILABLE(ios(13.0)) = 3
**eg. @(UIStatusBarStyleLightContent)
*/
@property (nonatomic,strong)NSNumber * yjdlShPreferredStatusBarStyle;
```

暗黑模式

```
/* UIUserInterfaceStyle
* UIUserInterfaceStyleUnspecified - 不指定样式，跟随系统设置进行展示
* UIUserInterfaceStyleLight - 明亮
* UIUserInterfaceStyleDark, - 暗黑 仅对iOS13+系统有效
*/
/*授权页 UIUserInterfaceStyle,默认:UIUserInterfaceStyleLight,eg. @(UIUserInterfaceStyleLight)*/
@property (nonatomic,strong) NSNumber * yjdlShAuthWindowOverrideUserInterfaceStyle;
```

封装动态配色工具类

```
+(UIColor *)generateDynamicColor:(UIColor *)lightModeColor darkModeColor:(UIColor *)darkModeColor {
if (@available(iOS 13.0, *)) {
return [UIColor colorWithDynamicProvider:\^UIColor * _Nonnull(UITraitCollection * _Nonnull traitCollection) {
if (traitCollection.userInterfaceStyle == UIUserInterfaceStyleDark) {
return darkModeColor;
} else {
return lightModeColor;
}
}];
}
```

```
return lightModeColor;

}
```

授权页面暗黑模式颜色适配

注：需先设置页面跟随系统模式显示

```
- (void)yjdlSh_login{

YjdlSUIConfigure*cfg = [YjdlSUIConfigurey jdlShDefaultUIConfigure];

cfg.viewController = self;

cfg.YjdlShOrientationLayOutPortrait = [YjdlShOrientationLayOut yjdlShDefaultOrientationLayOut];

// 开启跟随系统显示模式

if (@available(iOS 12.0, *)) {

cfg.yjdlShAuthWindowOverrideUserInterfaceStyle = @(UIUserInterfaceStyleUnspecified);

} else {

// Fallback on earlier versions

}

// 动态适配图片显示

cfg.yjdlShLogoImage = [UIImage imageNamed:@"apple_icon"];

// 动态适配背景色、文字颜色

cfg.yjdlShLoginBtnBgColor = [UIColorTools generateDynamicColor:UIColor.blackColor darkModeColor:UIColor.redColor];

cfg.yjdlShLoginBtnTextColor = [UIColorTools generateDynamicColor:UIColor.whiteColor
darkModeColor:UIColor.yellowColor];

cfg.customAreaView = \^(UIView * _Nonnull customAreaView) {

// 授权页面背景色动态配置

customAreaView.backgroundColor = [UIColorTools generateDynamicColor:UIColor.whiteColor
darkModeColor:UIColor.brownColor];

};

[YjdlShSDKManager quickAuthLoginWithConfigure:cfg openLoginAuthListener:\^(YjdlShCompleteResult * _Nonnull
completeResult) {

} oneKeyLoginListener:\^(YjdlShCompleteResult * _Nonnull completeResult) {

}];

}
```

授权页面暗黑模式图片适配

图片适配需要提供两套相同名字的图片，分别对应any-dark位置，如下图所示；

注意：

1.图片为同名，如"apple_icon@2x.png"(两张图片名称设置成一样),设置后图片会根据当前显示模式自动适配

2.右侧Appearances 需设置成“Any,Dark”模式**

图片适配示例代码

```
cfg.yjdlShLogoImage = [UIImage imageNamed:@"apple_icon"];
```

9.CheckBox勾选框位置调整

可调整属性

```
/*CheckBox

*协议勾选框，默认选中且在协议前显示

*可在sdk_oauth.bundle中替换checkBox_unSelected、checkBox_selected图片
```

*也可以通过属性设置选中和未选择图片

**/

/**协议勾选框（默认显示,放置在协议之前）BOOL eg.@(YES)*/

@property (nonatomic,strong) NSNumber *yjdlShCheckBoxHidden;

/**协议勾选框默认值（默认选中）BOOL eg.@(YES)*/

@property (nonatomic,strong) NSNumber *yjdlShCheckBoxValue;

/**协议勾选框 尺寸 NSValue->CGSize eg.[NSValue valueWithCGSize:CGSizeMake(25, 25)]*/

@property (nonatomic,strong) NSValue *yjdlShCheckBoxSize;

/**协议勾选框 UIButton.image图片缩进 UIEdgeInset eg.[NSValue valueWithUIEdgeInsets:UIEdgeInsetsMake(2, 2, 2, 2)]*/

@property (nonatomic,strong) NSValue *yjdlShCheckBoxImageEdgeInsets;

/**协议勾选框 设置CheckBox顶部与隐私协议控件顶部对齐 YES或大于0生效 eg.@(YES)*/

@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentToAppPrivacyTop;

/**协议勾选框 设置CheckBox对齐后的偏移量,相对于对齐后的中心距离在当前垂直方向上的偏移*/

@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentOffset;

/**协议勾选框 设置CheckBox顶部与隐私协议控件竖向中心对齐 YES或大于0生效 eg.@(YES)*/

@property (nonatomic,strong) NSNumber *yjdlShCheckBoxVerticalAlignmentToAppPrivacyCenterY;

/**协议勾选框 非选中状态图片*/

@property (nonatomic,strong) UIImage *yjdlShCheckBoxUncheckedImage;

/**协议勾选框 选中状态图片*/

@property (nonatomic,strong) UIImage *yjdlShCheckBoxCheckedImage;

CheckBox大小调整

直接设置size：yjdlShCheckBoxSize = [NSValue valueWithCGSize:CGSizeMake(25, 25)];

对齐方式（相对协议控件）

yjdlShCheckBoxVerticalAlignmentToAppPrivacyTop ： 顶部对齐

yjdlShCheckBoxVerticalAlignmentToAppPrivacyCenterY ：垂直中心对齐

勾选框图片位置微调

原理：CheckBox为标准UIButton控件，yjdlShCheckBoxImageEdgeInsets属性设置实际是进行[UIButton setImageEdgeInsets：];操作。

yjdlShCheckBoxImageEdgeInsets = [NSValue valueWithUIEdgeInsets:UIEdgeInsetsMake(2, 2, 2, 2)];

四.本机认证

**

1.初始化

同二、SDK使用说明**-->**初始化**

2.本机认证

方法原型

/**********本机认证(本机号码校验)**********/

+ (void)mobileCheckWithLocalPhoneNumberComplete:(YjdlShComplete)complete;

参数描述

参数	是否必填	类型	说明
complete	**必填**	YjdlShComplete	获取token回调，回调token信息

接口作用

**
本机号码校验 :验证指定手机号与本机SIM卡是否一致。(此接口仅返回token，手机号验证需调用服务端)

使用场景

- 在初始化接口后调用
- 输入手机号后进行校验

请求示例代码

ObjC**:

1. 导入SDK头文件 #import \<YjdlShSDK/YjdlShSDK.h>

```
[YjdlShSDKManager mobileCheckWithLocalPhoneNumberComplete:\^( YjdlShComplete* _Nonnull completeResult) {

if (completeResult.error) {

NSLog(@"mobileCheckWithLocalPhoneNumber:%@\n",completeResult.error.description)

}else{

NSLog(@"mobileCheckWithLocalPhoneNumber:%@\n",completeResult.yy_modelToJSONObject);

//调服务端接口，发送token，验证手机号

[weakSelf checkPhonenumber:completeResult.data];

}

}];

//此处模拟客户服务端调用服务端，依照服务端文档，实际情景下，app需要调用自己的服务端接口，将整个token传给服务端，整个token的验证在客户服务端进行，客户服务端将结果返回给app

- (void)checkPhonenumber:(NSDictionary *)completeResulyjdlShata{

NSLog(@"tokenParamr:%@",completeResulyjdlShata);

NSMutableDictionary * paramr = [NSMutableDictionary dictionaryWithDictionary:completeResulyjdlShata];

paramr[@"appId"] = APPID;

paramr[@"mobile"] = phoneNumberToCheck;

NSMutableString *formDataString = [NSMutableString new];

NSArray * keys = [paramr.allKeys sortedArrayUsingComparator:\^NSComparisonResult(id _Nonnull obj1, id _Nonnull obj2) {

NSString * objString1 = [NSString stringWithFormat:@"%@",obj1];

NSString * objString2 = [NSString stringWithFormat:@"%@",obj2];

return [objString1 compare:objString2];

}];

for (NSString * key in keys) {

[formDataString appendString:key];

[formDataString appendString:paramr[key]];

}

CocoaSecurityResult * hmacSha256Result = [CocoaSecurity hmacSha256:formDataString hmacKey:APPKEY];

paramr[@"sign"] = hmacSha256Result.hex;

NSLog(@"%@",paramr);

AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];

// 设置超时时间

[manager.requestSerializer willChangeValueForKey:@"timeoutInterval"];

manager.requestSerializer.timeoutInterval = 8.f;
```

```
[manager.requestSerializer didChangeValueForKey:@"timeoutInterval"];

[manager POST:yjdlSh_SDK_URL_MobileValidate parameters:paramr headers:nil progress:nil success:\^(NSURLSessionDataTask * _Nonnull task, id _Nullable responseObject) {

NSInteger code = [[responseObject valueForKey:@"code"] integerValue];

if (code == 200000) {

if ([responseObject[@"data"][@"isVerify"] integerValue] == 1) {

//本机校验成功，号码一致

if (completion) {

completion(@(YES),responseObject,nil);

}

}else{

//本机校验失败，号码不一致

if (completion) {

completion(@(NO),responseObject,nil);

}

}

}else{

//本机校验失败

if (completion) {

completion(nil,responseObject,nil);

}

}

} failure:\^(NSURLSessionDataTask * _Nullable task, NSError * _Nonnull error) {

if (completion) {

completion(nil,nil,error);

}

}];

}
```

3.手机号后台服务端校验

以下为代码块--完整方法请参考步骤 **2.本机认证**

```
- (void)getPhonenumber:(NSDictionary *)completeResulyjdlShata{

NSMutableDictionary * paramr = [NSMutableDictionary dictionaryWithDictionary:completeResulyjdlShata];

paramr[@"appId"] = APPID;

NSMutableString *formDataString = [NSMutableString new];

NSArray * keys = [paramr.allKeys sortedArrayUsingComparator:\^NSComparisonResult(id _Nonnull obj1, id _Nonnull obj2) {

NSString * objString1 = [NSString stringWithFormat:@"%@",obj1];

NSString * objString2 = [NSString stringWithFormat:@"%@",obj2];

return [objString1 compare:objString2];

}];

for (NSString * key in keys) {

[formDataString appendString:key];
```

```
[formDataString appendString:paramr[key]];

}

CocoaSecurityResult * hmacSha256Result = [CocoaSecurity hmacSha256:formDataString hmacKey:APPKEY];

paramr[@"sign"] = hmacSha256Result.hex;

NSLog(@"%@",paramr);

AFHTTPSessionManager *manager = [AFHTTPSessionManager manager];

// 设置超时时间

[manager.requestSerializer willChangeValueForKey:@"timeoutInterval"];

manager.requestSerializer.timeoutInterval = 8.f;

[manager.requestSerializer didChangeValueForKey:@"timeoutInterval"];

[manager POST:yjdlSh_SDK_URL_MobileQuery parameters:paramr headers:nil progress:nil success:\^(NSURLSessionDataTask *
_Nonnull task, id _Nullable responseObject) {

NSInteger code = [[responseObject valueForKey:@"code"] integerValue];

if (code == 200000) {

NSString * mobileName = responseObject[@"data"][@"mobileName"];

CocoaSecurityResult * appKey_md5_result = [CocoaSecurity md5:APPKEY];

NSString * appKey_md5 = appKey_md5_result.hexLower;

if (appKey_md5.length == 32) {

NSString * key = [appKey_md5 substringToIndex:16];

NSString * iv = [appKey_md5 substringFromIndex:16];

CocoaSecurityDecoder *decoder = [CocoaSecurityDecoder new];

CocoaSecurityResult *aes256Decrypt = [CocoaSecurity aesDecryptWithData:[decoder hex:mobileName] key:[key
dataUsingEncoding:NSUTF8StringEncoding] iv:[iv dataUsingEncoding:NSUTF8StringEncoding]];

NSString * mobileCode = aes256Decrypt.utf8String;

if (completion) {

completion(mobileCode,responseObject,nil);

}

}else {

if (completion) {

completion(nil,responseObject,nil);

}

}

} else {

if (completion) {

completion(nil,responseObject,nil);

}

}

} failure:\^(NSURLSessionDataTask * _Nullable task, NSError * _Nonnull error) {

if (completion) {

completion(nil,nil,error);

}
```

```
}];

}
```

五.返回码对照

外层错误码

同一外层码可能对应不同的内层码

外层返回码	返回码描述
1000	一键登录成功，解析result，可得到网络请求参数
1011	用户取消免密登录（点击返回按钮）
1001	SDK初始化失败
1023	预取号/取号失败
1003	拉起授权页失败/一键登录失败/获取token失败
1008	未开启移动网络
1032	账户禁用
2000	本机校验：获取token成功
2001	本机校验：手机号码为空
2003	本机校验：联通获取token失败
2004	本机校验：电信获取token失败
2005	本机校验：移动获取token失败
2009	本机校验：非三大运营商
2023	本机校验：未开启移动网络
其他	其他错误

内层错误码

通返回码

状态码	报错信息
100	成功(有数据返回)
111	认证失败(手机号码和认证手机号码不一致,计费)
112	认证失败(认证但未取到手机号码)
1101	公网 ip 无效(客户 wifi 访问、wap 方式访问等引起获取到的公网 ip 查询不到省份信息)
1102	私网 ip 无效(无法用私网 ip 找到对应的省份信息)
1103	待认证的手机号不能为空(认证传入的手机号码为空)
1104	授权码为空(传入的授权码为空)
1105	参数信息错误(参数名称、内容错误或者参数丢失再或者加密错误)
1106	应用密钥信息不匹配(密钥信息与包名不一致，android 还需要校验 MD5 信息，也有可能是参数命名错误)
1107	余额不足(使用条数不足)
1108	调用能力不匹配(取号置换码调用认证能力或者反之)
1201	取号失败
1202	认证失败
1203	获取置换码失败
2101	鉴权失败(参数 sign 名称错误或者 sign 值有误)
2102	accessCode 已失效(accessCode 错误或者过期)
2103	序列号不存在(序列号与授权码和密钥绑定，序列号不存在时返回，即授权码错误或者密钥信息与前台使用 不是一套，或者 seq 已过期)
2201	app_id 无效(appid 未激活或者 appid 错误)
2202	应用信息错误(获取应用信息错误(demo 中 data 参数中 app 信息错误))
2203	sdk 信息错误(获取 sdk 信息错误(demo 中 data 参数中 sdk 信息错误))
2205	接入信息解析错误(用户接入信息解析失败)
2206	流控值超限(用户访问流控超过限制)
3201	系统繁忙(服务端系统出现错误)
3202	内部网关错误
3203	内部路由错误
3204	无支付权限
3206	取号功能暂时不可用
3207	不支持此功能
10100	无网络连接
10101	无数据网络连接
10102	ApiKey 或 PublicKey 不能为空
10103	超时

10104	用户取消登录
10105	切换登录方式
10106	数据解密异常(SDK 解密数据失败)
10107	打开授权页
10110	取号中(正在取号中)
20100	测试次数超限
20101	10 分钟之内最多只能获取 30 个授权码
30200	服务端数据格式出错

电信返回码

返回码	返回描述
0	请求成功
-64	permission-denied(无权限访问)
-65	API-request-rates-Exceed-Limitations(调用接口超限)
-10001	取号失败
-10002	参数错误
-10003	解密失败
-10004	ip受限
-10005	异网取号回调参数异常
-10006	Mdn取号失败，且属于电信网络
-10007	重定向到异网取号
-10008	超过预设取号阈值
-10009	时间戳过期
-20005	sign-invalid(签名错误)
-20006	应用不存在
-20007	公钥数据不存在
-20100	内部解析错误
-20102	加密参数解析失败
-30001	时间戳非法
-30003	topClass 失效
51002	参数为空
51114	无法获取手机号
80000	请求超时
80001	请求网络异常
80002	响应码错误
80003	无网络连接
80004	移动网络未开启
80005	Socket 超时异常
80006	域名解析异常
80007	IO 异常
80008	No route to host
80009	nodename nor

移动返回码

返回码	返回码描述
103000	成功
103101	请求签名错误
103102	包签名/Bundle ID错误
103108	短信验证码错误
103109	短信验证码校验超时
103111	网关IP错误
103119	appid不存在
103125	短验下发时，手机号填写格式错误
103211	其他错误，（如有需要请联系qq群609994083内的移动认证开发）
103902	scrip失效
103911	token请求过于频繁，10分钟内获取token且未使用的数量不超过30个
103273	预取号联通重定向（暂不支持联通取号）
105002	移动取号失败
105003	电信取号失败
105021	已达当天取号限额
105302	appid不在白名单

105313	非法请求
200020	用户取消登录
200021	数据解析异常
200022	无网络
200023	请求超时
200025	其他错误（socket、系统未授权数据蜂窝权限等，如需要协助，请加入qq群发问）
200027	未开启数据网络
200028	网络请求出错
200038	异网取号网络请求失败
200048	用户未安装sim卡
200050	EOF异常
200060	切换账号（未使用SDK短验时返回）
200061	授权页面异常
200064	服务端返回数据异常
200072	CA根证书校验失败
200080	本机号码校验仅支持移动手机号
200082	服务器繁忙
200086	ppLocation为空
200087	授权页成功拉起
200089	SDK正在处理

七.已知问题

1.ATS开关(Http与Https)
目前运营商个别接口为http请求，对于全局禁用Http的项目，需要设置Http白名单。以下为运营商http接口host名单：

.cmpassport.com、*id6.me*、*123.125.99.8:9001*、*ms.zzx9.cn*、*mdn.open.wo.cn*、*10.99.255.231*，为通配符，建议按以下方式配置Info.plist

\<key>NSAppTransportSecurity\</key>

\<dict>

\<key>NSExceptionDomains\</key>

\<dict>

\<key>zzx9.cn\</key>

\<dict>

\<key>NSIncludesSubdomains\</key>

\<true/>

\<key>NSTemporaryExceptionAllowsInsecureHTTPLoads\</key>

\<true/>

\</dict>

\<key>cmpassport.com\</key>

\<dict>

\<key>NSIncludesSubdomains\</key>

\<true/>

\<key>NSTemporaryExceptionAllowsInsecureHTTPLoads\</key>

\<true/>

\</dict>

\<key>id6.me\</key>

\<dict>

\<key>NSIncludesSubdomains\</key>

\<true/>

\<key>NSTemporaryExceptionAllowsInsecureHTTPLoads\</key>

\<true/>

\</dict>

\<key>wostore.cn\</key>

\<dict>

\<key>NSIncludesSubdomains\</key>

\<true/>

\<key>NSTemporaryExceptionAllowsInsecureHTTPLoads\</key>

\<true/>

\</dict>

\<key>mdn.open.wo.cn\</key>

\<dict>

\<key>NSIncludesSubdomains\</key>

\<true/>

\<key>NSTemporaryExceptionAllowsInsecureHTTPLoads\</key>

\<true/>

\</dict>

\</dict>

\</dict>

Android版一键登录SDK开发文档

Android v2.3.3 一键登录SDK开发文档

下载SDK及相关文档

请在官网下载最新的SDK包（包含Demo+SDK+资源包）点击下载 非开发人员想体验demo，可直接下载apk包到手机 点击下载

.准备 作概述

合规性说明前 置 条 件 创建应

快速体验demo 开发环境搭建

. 键登录

.初始化

.预取号

.拉起授权

.销毁授权

.授权 点击事件监听

.其他API

.置换 机号

.授权 界 修改a.设计规范

b.授权 配置

c.添加 定义控件d.设置弹窗样式e.设置横竖屏

三.本机认证

.初始化

.本机号校验

.校验 机号四.返回码

.准备 作

概述

本 是 键登录SDK_Android v2.3.3版本的接 档， 于指导SDK的使 法，默认读者已经熟悉 IDE（Eclipse 或者 Android Studio）的基本使 法，以及具有 定的 Android 编程知识基础。

合规性说明

为了保证您的App顺利通过检测，我们制作了Android统计SDK初始化合规 案。合规三步 ：

1、您需要确保App有《隐私政策》，并且在 户 次启动App时就弹出《隐私政策》取得 户同意。

2、您务必告知 户您选择 键登录SDK服务，请在《隐私政策》中增加如下参考条款：

“我们的产品为改善注册及登录界 户体验，集成第三 SDK 键登录服务： 键登录SDK， 于帮助实现 户 键登录 为。为了实现 关取号技术， 键登录SDK需要获取IP地址、 卡（MAC）地

址、国际移动设备识别码（IMEI）、OAID（替代IMEI）、sim卡信息，并会收集 机机型、系统类型、系统版本、 络环境、 关取号报错 志数据以提供统计分析服务能 ，并可提供反欺诈等功能。

3、您务必确保 户同意《隐私政策》之后，再初始化 键登录SDK。具体初始化步骤详 下 。

前置条件

- 键登录SDK 持minSdkVersion 16及以上版本
- 键登录SDK 持中国移动3/4G/5G、联通3/4G/5G、电信4G/5G的取号能 ，在3G 络下时延会更
- 键登录SDK 持单数据 络/数据 络与WiFi 络双开，不 持单WiFi 络
- 对于双卡 机， 键登录SDK只对当前流量卡取号，双卡均未开数据流量SDK将会返回错误码

本机号码校验使 场景：

户输 机号码，通过SDK获取token（通过authenticationRespond获取），服务端携带输 的 机号码和token

去运营商 关进 校验（服务端调 mobile-validate接 ），返回的结果时 户当前上 使 的号码与 输 的号码是否 致。

键登录使 场景：

户 需输 机号码，SDK会拉起授权 ， 户确认授权后，SDK会获取token（通过getOneKeyLoginStatus中获

取），服务端携带token到运营商 关获取 户当前上 使 的号码（服务端调 mobile-query接

），并返回给APP服务端。

创建应

应 的创建流程及APPID的获取，请查看「账号创建」 档

注意：如果应 有多个包名或签名不同的 甲包，须创建多个对应包名和签名的应 ，否则 甲包将报包名或签名校验不通过。

快速体验demo

- Android压缩包附带的apk 件夹中是 键登录demo的安装包，可以直接安装到Android 机上。并快速体验 键登录在您的 机上的表现。
- Android压缩包附带的demo 件夹中是 键登录的示例 程，使 Android studio打开示例 程， 完成以下步骤配置，然后直接运起来测试。

  a.将build 的applicationId换成对应的测试包名b.将build 的签名配置改成您的签名配置

  c.将初始化appid换成您在 键登录平台创建应 后 成的appid

开发环境搭建

1. 将开发包拷 到 程
2. 将SDK中libs 录下的aar包拷 到您 程的libs 录下，如没有该 录需新建。

   并在build 件的dependencies中添加aar包依赖：

1. SDK中jniLibs 录下的so库复制到您 程的jniLibs 录下，如没有该 录需新建。

   jniLibs所在 录结构如下图：

注意：系统默认读取jniLibs 录下的so库，如果您的项 中在build 指定了 jniLibs.srcDirs = ['libs']，则必须将so库放到libs 录下，否则so库会引 不到，导致电信卡预取号报："result":80102,"msg":"预登录异常"。

SDK提供了多个架构的so库，可根据项 需求选择相应的so库。如果您的项 只包含某个 录，则复制对应 录的so库，例如，您的项 中只有armeabi-v7a 录，则只复制jniLibs中的armeabi-v7a 录下的so库到您的项 。

1. 配置AndroidManifest.xml 件

必要权限：

建议的权限：如果选 该权限，需要在预取号步骤前提前动态申请。

建议开发者申请本权限，本权限只 于移动运营商在双卡情况下，更精准的获取数据流量卡的运营商类 型，缺少该权限，存在取号失败概率上升的 险。

配置权限说明

权限名称	权限说明	使 说明
INTERNET	允许应 程序联 于访问 关和认证服务器	
ACCESS_WIFI_STATE	允许访问WiFi 络状态信息	允许程序访问WiFi 络状态信息
ACCESS_NETWORK_STA TE	允许访问 络状态	区分移动 络或WiFi 络
CHANGE_NETWORK_STA TE	允许改变 络连接状态	设备在WiFi跟数据双开时， 强 切换使 数据 络
CHANGE_WIFI_STATE	允许改变WiFi 络连接状态	设备在WiFi跟数据双开时， 强 切换使
READ_PHONE_STATE	允许读取 机状态	（可选）获取IMSI 于判断双卡和换卡
WRITE_SETTINGS	允许读写系统设置项	电信SDK在6.0系统以下进 数据切换 到的权限，添加后可增加电信在WiFi+4G下 络环境下的取号成功率。6.0系统以上可去除
GET_TASKS	允许应 程序访问TASK	

在application标签内配置授权登录activity，screenOrientation和theme可以根据项 需求 修改

1. \<activity
   1. android:name="com.shlogin.sdk.view.CmccLoginActivity"

      XML

      复制代码
2. android:configChanges="keyboardHidden|orientation|screenSize"
   1. android:launchMode="singleTop" /> 5
3. \<activity-alias
4. android:name="com.cmic.sso.sdk.view.LoginAuthActivity"
5. android:configChanges="keyboardHidden|orientation|screenSize"
6. android:launchMode="singleTop"
7. android:targetActivity="com.shlogin.sdk.view.CmccLoginActivity"
8. android:theme="@style/dialogStyle" />
9. \<activity

```
10. android:name="com.shlogin.sdk.view.OneKeyLoginActivity"
11. android:configChanges="keyboardHidden|orientation|screenSize"
12. android:launchMode="singleTop"
13. android:screenOrientation="behind" 17 />
14. \<activity
15. android:name="com.shlogin.sdk.view.PrivacyProtocolActivity"
16. android:configChanges="keyboardHidden|orientation|screenSize"
17. android:launchMode="singleTop"
18. android:screenOrientation="behind" 23 />
```

配置Android9.0对http协议的 持两种 式：

式 ：

示例代码：

式 ：

运营商个别接 为http请求，对于全局禁 Http的项 ，需要设置Http 名单。以下为运营商http接 域名：

cmpassport.com；*.10010.com，119.3.251.136

混淆规则：

1. AndResGuard资源压缩过滤：

通过上 的 个步骤， 程就配置完成了，接下来就可以在 程中使 键登录SDK进 开发了。

. 键登录

初始化

调 SDK其他任何 法前，请确保已调 过初始化，否则会失败并返回未初始化。

法原型

参数描述

参数	类型	说明
context	Context	传ApplicationContext对象
appId	String	应 的appId
initListener	InitListener	初始化回调监听，getInitStatu是该监听唯 的抽象 法，即void getInitStatus(int code, String result)

getInitStatus(int code, String result) 法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为1022:成功；其他：失败
result	String	返回信息

注意：初始化不要放在申请权限 法内，否则 户拒绝权限会造成后续 法没有回调。

1. 预取号

建议在判断当前 户属于未登录状态时使 ，已登录状态 户请不要调 该 法

- 建议在执 拉取授权登录 的 法前，提前 段时间调 预取号 法，中间最好有2-3秒的缓冲

  （因为预取号

  法需要1\~3s的时间取得临时凭证）， 如放在启动 的onCreate（） 法中，或者app启动的

  application中的

  onCreate（） 法中去调 ，不建议放在 户登录时和拉取授权登录 法 起调 ，会影响 户体验和成功 率。

请勿频繁的多次调 、请勿与拉起授权登录 同时和之后调 。

- 避免 量资源下载时调 ，例如游戏中加载资源或者更新补丁的时候要顺序执

法原型：

参数描述

参数	类型	说明
getPhoneInfoListen er	GetPhoneInf oListener	预取号回调监听，getPhoneInfoStatus是该监听中唯 的抽象 法，即void getPhoneInfoStatus(int code, String result)

法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	Int	code为1022：成功；其他：失败
result	String	返回信息

1. 拉起授权

调 拉起授权 法后将会调起运营商授权 。已登录状态请勿调 。

- 每次调 拉起授权 法前均需先调 授权 配置 法，否则授权 可能会展示异常。
  - 1秒之内只能调 次， 且必须保证上 次拉起的授权 已经销毁再调 ，否则SDK会返回请求频 繁。

法原型：

参数描述

字段	类型	含义
isFinish	boolean	点击授权 键登录按钮有回调时是否 动销毁授权 ： true： 动销毁 false:不 动销毁，开发者需主动调 销毁授权 法进 授权 销毁操作
openLoginAu thListener	OpenLoginAu thListener	授权 是否拉起成功监听，getOpenLoginAuthStatus是该接 中唯 的抽象 法，即 void getOpenLoginAuthStatus(int code,String result)
oneKeyLogin Listener	OneKeyLogin Listener	点击授权 登录按钮及返回键监听，getPhoneCode是该接 中唯 的抽象 法，即void getOneKeyLoginStatus(int code,String result) （code等于1011为点击返回键（包括物理返回键），其他均为点击 键 登录按钮的回调）

getOpenLoginAuthStatus(int code,String result) 法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为1000：授权 成功拉起其他：失败
result	String	返回信息

getOneKeyLoginStatus(int code,String result) 法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为1000：成功 其他：失败（包含点击返回键 code=1011）
result	String	返回信息

含义如下：

字段	类型	含义
token	String	来和后台置换 机号。 次有效。

1. 销毁授权
2. 授权 动销毁
    1. 在授权登录 ，当 户主动点击左左上 返回按钮时，返回码为1011，SDK将 动销毁授权 ；
    2. 安卓 SDK，当 户点击 机的硬件返回键（相当于取消登录），返回码为1011，SDK将 动销毁授权

1. 当 户设置 键登录或者其他 定义控件为 动销毁时，得到回调后，授权 会 动销毁
2. 授权 动销毁

当设置 键登录为 动销毁时，点击授权 键登录按钮成功获取token不会 动销毁授权 ，请务必在回调中处理完 的逻辑后 动调 销毁授权 法。

1. 当设置 定义控件为 动销毁时，请务必在回调中处理完 的逻辑后 动调 销毁授权 法。

法原型

示例代码

注意：点击授权 的返回按钮或者 定义控件时，请不要回调或者返回到的界 主线程中做耗时操作， 避免堵塞授权 销毁，导致再次调 拉起授权 时返回请求频繁。

1. 授权 点击事件监听

需要对授权 点击事件监听的 户，可调 此 法监听授权 点击事件， 此需求可以不写。

法原型

参数描述

参数	类型	说明
actionListener	ActionListener	点击事件回调监听，ActionListener是该监听唯 的抽象 法，即 void ActionListener(int code, String result,String operator)

ActionListners(int type,int code,String message) 法返回参数含义如下：

字段	类型	含义
type	int	type=1 ，隐私协议点击事件type=2 ，checkbox点击事件type=3 ， 键登录按钮点击事件
code	int	type=1 ，隐私协议点击事件，code分为0,1,2,3（协议 序号） type=2 ，checkbox点击事件，code分为0,1（0为未选中，1 为选中） type=3 ， 键登录点击事件，code分为0,1（0为协议未勾选时，1为协议勾选时）
message	String	点击事件的详细信息

1. 其他API
2. 设置授权 loading显示隐藏（授权 拉起之后调 ）

法原型

参数描述

参数	类型	说明
visibility	boolean	点击 键登录后，授权 展示的loading view（设置true 显示； false隐藏）

1. 设置授权 协议复选框是否选中（授权 拉起之后调 ）

法原型

参数描述

参数	类型	说明
isChecked	boolean	设置协议复选框是否选中（设置true 选中；false 未选中）

1. 置换 机号

当 键登录外层code为1000时，您将获取到返回的参数，请将这些参数传递给后端开发 员，并参考

「服务端」 档来实现获取 机号码的步骤。

1. 授权 界 修改
2. 设计规范

注意：开发者不得通过任何技术 段，将授权 的隐私栏、品牌露出内容隐藏、覆盖，对于接 键 登录SDK并上线的应 ，我 和运营商会对授权 做审查，如果有出现未按要求设计授权 ，我 有权将应 的登录功能下线。

1. 授权 配置

调 该 法可实现对三 运营商授权 个性化设计，每次调 拉起授权 法前必须先调 该 法， 否则授权界

会展示异常，具体实现可参考demo示例的ConfigUtils配置类。（三 界 配置内部实现逻辑不同，请务必使 移

动、联通、电信卡分别测试三 界 ）

法原型

参数说明

参数	参数类型	说明
portraitYanUICo nfig	LoginUIConfig	竖屏 样式配置对象，开发者在 LoginUIConfig.java类中调 对应的 法配置授权 中对应的元素（该参数不能传null， 否则展示默认 ）
landYanUIConfig	LoginUIConfig	横屏 样式主题配置对象，开发者在 LoginUIConfig.java类中调 对应的 法配置授权 中对应的元素（针对指定屏幕 向的，该参数传null）

LoginUIConfig.java配置元素说明

授权 背景配置三选 ， 持图 ，gif图，视频

法	参数类型	说明
setAuthBGImgPath	Drawable	设置授权 背景图
setAuthBgGifPath	String	只 持本地gif图，需要放置到drawable 件夹中。传 图 名称即可。
setAuthBgVideoPath	String	持本地路径如："android.resource://" + context.getPackageName() + "/" + R.raw.testvideo" 持 络路径："https://xxx"

设置授权 进出场动画

法	参数类型	说明
setActivityTranslateAnim	string	参数1：进场动画传xml 件名即可参数2：退场动画传xml 件名即可

授权 状态栏

法	参数类型	说明
setStatusBarColor	int	设置状态栏背景颜
setLightColor	boolean	设置状态栏字体颜 是否为

setStatusBarHidden	boolean	设置状态栏是否隐藏
setVirtualKeyTranspar ent	boolean	设置虚拟键是否透明

授权 字体

法	参数类型	说明
setTextSizeIsdp	boolean	设置字体是否以dp为单位

授权 导航栏

法	参数类型	说明
setFullScreen	boolean	设置是否全屏显示（true：全屏；false：不全屏）默认不全屏
setNavColor	int	设置导航栏背景颜 （默认透明）
setNavText	string	设置导航栏标题 字
setNavTextColor	int	设置导航栏标题 字颜
setNavTextSize	int（单位sp）	设置导航栏标题 字
setNavReturnImgPath	Drawable	设置导航栏返回按钮图标
setNavReturnImgHidd en	boolean	设置导航栏返回按钮是否隐藏（true：隐藏；false：不隐藏）
setNavReturnBtnWidth	int（单位dp）	设置导航栏返回按钮宽度（默认25dp）
setNavReturnBtnHeigh t	int（单位dp）	设置导航栏返回按钮 度（默认25dp）
setNavReturnBtnOffse tRightX	int（单位dp）	设置导航栏返回按钮距离屏幕右侧X偏移
setNavReturnBtnOffse tX	int（单位dp）	设置导航栏返回按钮距离屏幕左侧X偏移
setNavReturnBtnOffse tY	int（单位dp）	设置导航栏返回按钮距离屏幕上侧Y偏移
setAuthNavHidden	boolean	设置导航栏是否隐藏（true：隐藏；false：不隐藏）
setAuthNavTransparen t	boolean	设置导航栏是否透明（true：透明；false：不透明）
setNavTextBold	boolean	设置导航栏字体是否加粗（true：加粗；false：不加粗）

授权 logo

法	参数类型	说明
setLogoImgPath	Drawable	设置logo图
setLogoWidth	int（单位dp）	设置logo宽度
setLogoHeight	int（单位dp）	设置logo 度
setLogoOffsetY	int（单位dp）	设置logo相对于标题栏下边缘y偏移
setLogoOffsetBottom Y	int（单位dp）	设置logo相对于屏幕底部y偏移
setLogoHidden	boolean	设置logo是否隐藏（true：隐藏；false：不隐藏）
setLogoOffsetX	int（单位dp）	设置logo相对屏幕左侧X偏移

授权 号码栏

法	参数类型	说明
setNumberColor	int	设置号码栏字体颜
setNumFieldOffsetY	int（单位dp）	设置号码栏相对于标题栏下边缘y偏移
setNumFieldOffsetBot tomY	int（单位dp）	设置号码栏相对于屏幕底部y偏移
setNumFieldWidth	int（单位dp）	设置号码栏宽度
setNumFieldHeight	int（单位dp）	设置号码栏 度
setNumberSize	int(单位sp)	设置号码栏字体
setNumFieldOffsetX	int（单位dp）	设置号码栏相对屏幕左侧X偏移
setNumberBold	boolean	设置号码栏字体是否加粗（true：加粗；false：不加粗）

授权 登录按钮

法	参数类型	说明
setLogBtnText	string	设置登录按钮 字
setLogBtnTextColor	int	设置登录按钮 字颜
setLogBtnImgPath	Drawable	设置授权登录按钮图
setLogBtnOffsetY	int（单位dp）	设置登录按钮相对于标题栏下边缘Y偏移
setLogBtnOffsetBotto mY	int（单位dp）	设置登录按钮相对于屏幕底部Y偏移
setLogBtnTextSize	int（单位sp）	设置登录按钮字体
setLogBtnHeight	int（单位dp）	设置登录按钮 度
setLogBtnWidth	int（单位dp）	设置登录按钮宽度
setLogBtnOffsetX	int（单位dp）	设置登录按钮相对屏幕左侧X偏移
setLogBtnTextBold	boolean	设置登录按钮字体是否加粗（true：加粗；false：不加粗）

授权 隐私栏

法	参数类型	说明
setAppPrivacyOne	string（链接）	设置开发者隐私条款1名称和URL 参数1：名称 参数2：url
setAppPrivacyTwo	string（链接）	设置开发者隐私条款2名称和URL 参数1：名称 参数2：url
setAppPrivacyThree	string（链接）	设置开发者隐私条款3名称和URL 参数1：名称 参数2：url
setPrivacySmhHidden	boolean	设置协议名称是否显示书名号《》，默认显示书名号（true：不显示；false：显示）
setPrivacyTextSize	int（单位sp）	设置隐私栏字体
setAppPrivacyColor	int	设置隐私条款名称颜 参数1：基础 字颜 参数2 ：协议 字颜
setPrivacyOffsetBotto mY	int（单位dp）	设置隐私条款相对于授权 底部下边缘y偏移
setPrivacyOffsetY	int（单位dp）	设置隐私条款相对于授权 标题栏下边缘y偏移
setPrivacyOffsetX	int（单位dp）	设置隐私条款相对屏幕左侧X偏移
setPrivacyOffsetGravit y Left	boolean	设置隐私条款 字多 显示时是否左对 （true：左对 ；false：居中）
setPrivacyState	boolean	设置隐私条款的CheckBox是否选中（true：选中； false：未选中）
setUncheckedImgPath	Drawable	设置隐私条款的CheckBox未选中时图
setCheckedImgPath	Drawable	设置隐私条款的CheckBox选中时图
setCheckBoxHidden	boolean	设置隐私条款的CheckBox是否隐藏（true：隐藏； false：不隐藏）
setCheckBoxWH	int（单位dp）	设置checkbox的宽 ，包含两个参数：1.宽 2. （默认宽 13dp）
setcheckBoxOffsetXY （2.3.1.6及之后版本新增 法）	int（单位dp）	设置checkbox在协议框 控件中的位置，包含两个参数：1.左偏移量 2.上偏移量（不设置默认居中）
setCheckBoxMargin	int（单位dp）	设置checkbox热点局域范围，包含四个参数：1.向左扩 2.向上扩 3.向右扩 4.向下扩（默认各10dp）
setPrivacyText	String	设置隐私条款名称外的 字，包含五个参数，参数示例 （红 字体为传 参数）： 同意《**》和《***》、《***》、《***》并授权获取本机号码 参数1：同意参数2：和参数3：、 参数4：、参数5：并授权获取本机号码
setPrivacyTextBold	boolean	设置协议栏字体是否加粗（true：加粗；false：不加粗）
setPrivacyCustomToas t	Toast	未勾选时， 定义点击 键登录的Toast提示
setPrivacyCustomToas tText	String	未勾选协议时toast提示 字
setPrivacyNameUnderl ine	boolean	协议是否显示下划线（true：显示；false：不显示）
setOperatorPrivacyAtL ast	boolean	运营商协议是否为最后 个显示（true：最后显示； false：不在最后显示）
setPrivacyGravityHoriz ontalCenter	boolean	设置隐私协议栏是否居中显示（true：居中；false：居左）

授权 slogan（**提供认证服务）

法	参数类型	说明
setSloganTextColor	int	设置slogan 字颜
setSloganTextSize	int（单位sp）	设置slogan 字字体
setSloganOffsetY	int（单位dp）	设置slogan相对于标题栏下边缘y偏移
setSloganHidden	boolean	设置slogan是否隐藏（true：隐藏；false：不隐藏）
setSloganOffsetBottom Y	int（单位dp）	设置slogan相对屏幕底部Y偏移
setSloganOffsetX	int（单位dp）	设置slogan相对屏幕左侧X偏移
setSloganTextBold	boolean	设置slogan 字字体是否加粗（true：加粗；false： 不加粗）

授权 loading

法	参数类型	说明
setLoadingView	ViewGroup	设置 定义loading

授权 相对控件设置（指定在登录按钮和协议栏之间）

法	参数类型	说明
setRelativeCustomVie w	View	设置 定义布局
	boolean	点击布局是否需要销毁授权 ：true销毁 false不销毁
	int（单位dp）	布局左间距
	int（单位dp）	距登录按钮的上边距
	int（单位dp）	布局右边距
	int（单位dp）	距协议栏的下边距
	CustomInterfac e	整个view的点击事件，内部的 view点击事件需 实现

协议 导航栏

法	参数类型	说明
setPrivacyActivityTranslateAn im	String	参数1：进场动画传xml 件名即可参数2：退场动画传xml 件名即可
setPrivacyStatusBarHidden	boolean	协议 是否隐藏状态栏
setPrivacyStatusBarColor	int	协议 状态栏颜
setPrivacyLightColor	boolean	协议 状态栏 字颜 是否为亮
setPrivacyTitleArray	String[]	协议 标题名称，不设置默认展示授权 对应的协议名称
setPrivacyNavColor	int	设置协议 导航栏背景颜
setPrivacyNavTextBold	boolean	设置协议 导航栏标题 字是否加粗（true：加粗；false：不加粗）
setPrivacyNavTextColor	int	设置协议 导航栏标题 字颜
setPrivacyNavTextSize	int（单位sp）	设置协议 导航栏标题 字
setPrivacyNavReturnImgPath	Drawable	设置协议 导航栏返回按钮图标
setPrivacyNavReturnImgHidd en	boolean	设置协议 导航栏返回按钮是否隐藏（true：隐藏；false：不隐藏）
setPrivacyNavReturnBtnWidt h	int（单位dp）	设置协议 导航栏返回按钮宽度
setPrivacyNavReturnBtnHeig ht	int（单位dp）	设置协议 导航栏返回按钮 度
setPrivacyNavReturnBtnOffse tRightX	int（单位dp）	设置协议 导航栏返回按钮距离屏幕右侧X偏移
setPrivacyNavReturnBtnOffse tX	int（单位dp）	设置协议 导航栏返回按钮距离屏幕左侧X偏移
setPrivacyNavReturnBtnOffse tY	int（单位dp）	设置协议 导航栏返回按钮距离屏幕上侧Y偏移

授权 遮盖层

法	参数类型	说明
setDialogDimAmount	float	授权 遮盖层灰度设置范围是（0\~1）

隐私协议提示弹框

法	参数类型	说明
addCustomPrivacyAlertVie w	view	添加授权 上显示隐私协议弹框

a.控件X偏移如果不设置默认居中。

添加 定义控件

调 该 法可实现在授权 添加 定义控件。

法原型

参数说明

参数	参数类型	说明
view（必填）	View	定义控件对象
isFinish（必填）	boolean	是否需要销毁授权 ：true销毁 false不销毁
type（必填）	boolean	设置 定义控件的位置： true为授权 导航栏 false为授权 导航栏以下空 处
customInterface	CustomInterface	定义控件监听

1. 设置弹窗样式

调 该 法可实现将授权 设置成弹窗样式。

法原型

参数说明

参数	参数类型	说明
isdialogTheme（必填）	boolean	是否 弹窗样式：true 弹窗样式 false 弹窗样式
dialogWidth（必填）	int（单位dp）	授权 弹窗宽度
dialogHeight（必填）	int（单位dp）	授权 弹窗 度
dialogX	int（单位dp）	授权 弹窗X偏移量（以屏幕中 为原点）

dialogY	int（单位dp）	授权	弹窗Y偏移量（以屏幕中 为原点）
isDialogBottom	boolean	授权	弹窗是否贴于屏幕底部： true：显示到屏幕底部，dialogY参数设置将 效false：不显示到屏幕底部，以dialogY参数为准

注意：设置弹窗效果背景的透明度需要在Manifest.xml 配置授权界 样式样式示例：

1. 为授权界 的activity设置弹窗theme主题
2. \<activity
    1. android:name="com.shlogin.sdk.view.CmccLoginActivity"

       Java

       复制代码
3. android:configChanges="keyboardHidden|orientation|screenSize"
    1. android:launchMode="singleTop"
    2. android:screenOrientation="behind"
    3. android:theme="@style/Theme.ActivityDialogStyle" 7 />

       8

       9 \<activity-alias
4. android:name="com.cmic.sso.sdk.activity.LoginAuthActivity"
5. android:configChanges="keyboardHidden|orientation|screenSize"
6. android:launchMode="singleTop"
7. android:screenOrientation="behind"
8. android:targetActivity="com.shlogin.sdk.view.CmccLoginActivity"
9. android:theme="@style/Theme.ActivityDialogStyle" 16 />
10. \<activity
11. android:name="com.shlogin.sdk.view.OneKeyLoginActivity"
12. android:configChanges="keyboardHidden|orientation|screenSize"
13. android:launchMode="singleTop"
14. android:screenOrientation="behind"
15. android:theme="@style/Theme.ActivityDialogStyle" 23 />
16. 设置theme主题的style样式

注意：如果需要触摸弹窗外部销毁授权 ，style的parent请使 系统dialog相关主题。

设置横竖屏

在manifest 件中，指定授权 activity的screenOrientation即可

注意：只有全屏不透明的activity才能指定 向，否则在8.0系统版本上会报 Only fullscreen opaque

activities can request orientation

即：弹框或者透明主题，授权 不能指定 向。如需指定 向，可以指定授权 前个 的 向，授权

设置跟随前个界 向，即： android:screenOrientation="behind"

三.本机认证

注：本机认证同免密登录，需要初始化，本机认证、免密登录可共 初始化，两个功能同时使 时，只需调 次初始化即可。

1. 初始化

同SDK使 说明-->初始化

本机号校验

在初始化执 之后调 ，本机号校验界 需 实现，可以在多个需要校验的 中调 。

法原型：

参数描述：

参数	类型	说明
authenticationExecuteListener（必填）	AuthenticationExecuteListener	本机号校验回调监听器，需要调 者 实现；是接 中的本机号校验参数回调接 ，authenticationRespond是该接 中唯 的抽象 法，即 void authenticationRespond(int code,String result)

示例代码：

authenticationRespond(int code,String result) 法返回参数分为外层code和result，含义如下：

字段	类型	含义
code	int	code为1000：成功其他：失败
result	String	返回信息

当外层code为1000时，result的返回为

含义如下：

字段	类型	含义
token	String	来和后台校验 机号。 次有效。

1. 校验 机号

当本机号校验外层code为2000时，您将获取到返回的参数，请将这些参数传递给后端开发 员，并参 考「服务端」 档来实现校验本机号的步骤

四.返回码

返回码	返回码描述
1000	键登录获取token成功
1001	运营商通道关闭
1002	运营商信息获取失败，请结合result查看具体失败原因
1003	键登录获取token失败，请结合result查看具体失败原因
1007	络请求失败，请结合result查看具体失败原因
1011	点击返回， 户取消免密登录
1014	SDK内部异常，请结合result查看具体失败原因
1016	APPID为空
1019	其他错误，请结合result查看具体失败原因
1022	络初始化、预取号成功
1023	初始化、预取号失败，请结合result查看具体失败原因
1031	请求过于频繁
1032	户禁
1000	本机号校验获取token成功
2003	本机号校验返回失败，请结合result查看具体失败原因

特殊说明：

关于接口是否用蜂窝网络调用

1.移动卡：预取号成功过，以后都可以纯wifi取号，不限次数

2.联通、电信卡：预取号成功过，只有第一次可以纯WiFi取号，取号成功过一次之后预取号就失效了

如果想每次都相同，可以在清理预取号缓存；

OneKeyLoginManager.getInstance().clearScripCache(getApplicationContext())