



一键登录

iOS SDK 接入文档

天翼云科技有限公司

修订记录：

版本号	发布时间	更新内容
V1.2.6	20230419	1, 认证品牌文本自定义 2, 新增授权页属性 3, 新增添加二次弹窗功能
V1.2.5	20231228	4, 新增本机校验超时时间 5, 新增获取 token 超时时间 6, 新增删除预取号信息接口 7, 新增授权页属性
V1.2.4	20231130	1, 新增设置预取号超时时间 2, 新增授权页关闭完成回调 3, 支持横屏授权页 4, 新增全屏授权页进出动画配置 5, 新增授权页属性
V1.2.3	20230920	1, 修复已知 bug（适配 16.4）
V1.2.2	20230809	1, 适配弹窗高度 2, 修复已知 bug
V1.2.1	20230630	1, 新增获取运营商类型接口 2, 新增底部弹窗
V1.2.0	20230531	1, 新增登录按钮点击事件回调
V1.1.3	20230331	1, 优化获取 token 次数
V1.1.2	20230301	1, 支持掩码手机号加粗 2, 支持默认隐私协议书名号配置 3, 新增获取 sdk 版本号接口 4, 新增控制台日志输出开关接口 5, 新增判断设备移动网络是否开启接口
v1.1.1	20230209	1, 新增网关配置能力 2, 修改已知 bug
v1.1.0	20221230	1, 新增本机号码校验
v1.0.4	20221130	1, 新增关闭按钮回调 2, 自定义协议支持多个 3, 新增关闭授权页方法 4, 支持勾选监听框状态
v1.0.3	20221104	1, 初始化接口新增验签
v1.0.2	20220920	1, 支持修改弹窗大小 2, 支持添加自定义 iView 3, 控件位置支持 X 轴设置
v1.0.1	20220815	1, 添加其它方式登录回调
V1.0.0	20220630	1, 发布 iOS 一键登录基本功能

目录

1	<i>iOS SDK 接入指南</i>	4
1.1	注意事项	4
2	<i>iOS SDK 使用方法</i>	4
2.1	环境配置	4
2.2	初始化接口	7
2.3	预取号接口	8
2.4	调起授权页接口	9
2.5	点击其它登录方式回调	10
2.6	点击关闭授权页按钮回调	11
2.7	点击授权登录未同意隐私协议回调	11
2.8	点击隐私协议勾选框回调	11
2.9	开启二次弹窗后，未勾选隐私协议时点击登录按钮的回调	11
2.10	点击登录按钮回调	11
2.11	授权页关闭完成回调	12
2.12	二次弹窗获取 token，仅在开启二次弹窗后生效	12
2.13	获取手机号能力	12
2.14	本机号码校验	12
2.15	判断设备移动数据网络是否开启	13
2.16	控制台日志输出开关	13
2.17	获取 sdk 版本号	13
2.18	删除预取号信息	13
2.19	获取运营商类型	13
2.20	授权页面自定义 UI 参数	14

1 iOS SDK 接入指南

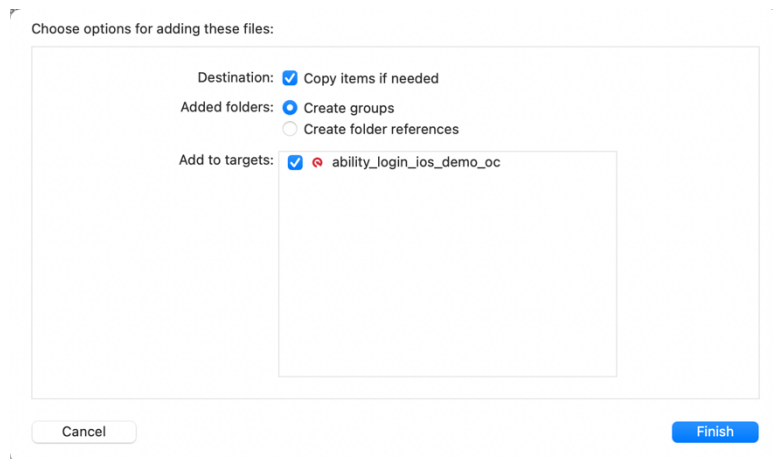
1.1 注意事项

1. 使用一键登录服务用户须打开蜂窝数据流量，取号过程中会消耗用户少量的流量。
2. sdk 中自带有授权页面的 UI 逻辑，开发者可通过 SDK 接口暴露参数修改部分 UI 界面样式。
3. 如果用户使用双卡手机，sdk 会使用当前设备上网的流量就行预取号操作。
4. 建议发版测试 移动、联通、电信 三家运营商取号功能

2 iOS SDK 使用方法

2.1 环境配置

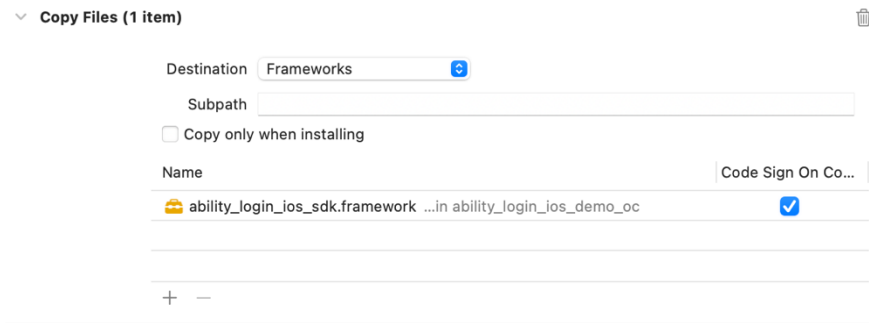
1. 把 ability_login_ios_sdk.framework 拖入工程中，勾选 Copy item if needed。



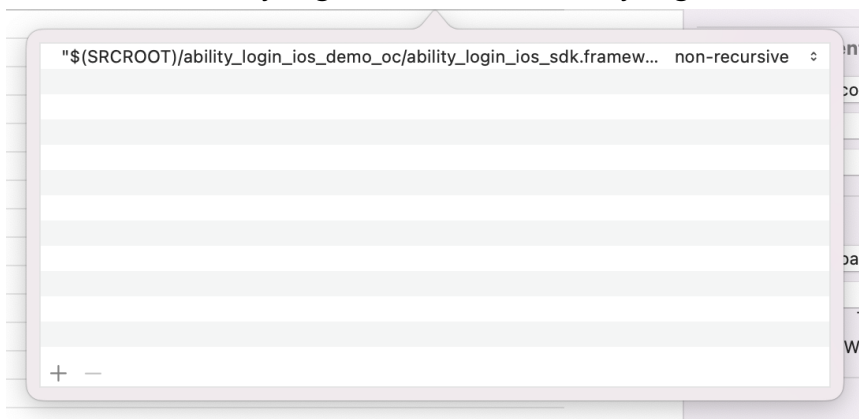
2. 把 AbilityResource.bundle 资源包拖入工程，勾选 Copy item if needed。
3. info.plist 文件中添加 App Transport Security Settings / Allow Arbitrary Loads 子项设置为 YES。

Key	Type	Value
Information Property List	Dictionary	(2 items)
App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Application Scene Manifest	Dictionary	(2 items)

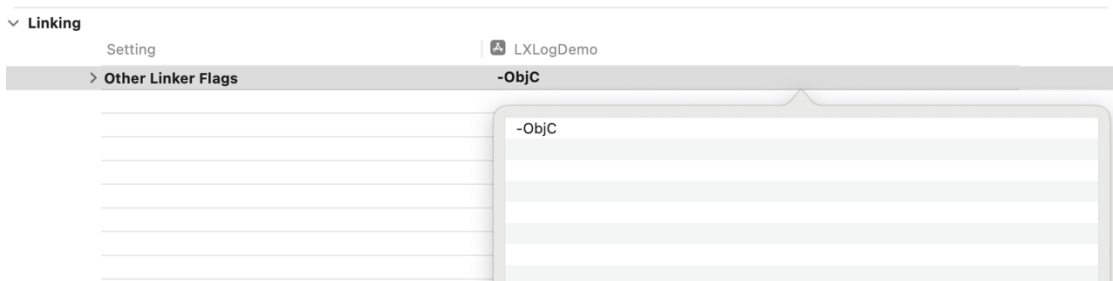
4. build phases 里添加 Copy Files, Destination 设置为 Frameworks , 点击+ , 添加 ability_login_ios_sdk.framework



5. build settings 里设置 Header Search Paths 设置 ability_login_ios_sdk.framework 文件的路径，设置为
"\${SRCROOT}/ability_login_ios_demo_oc/ability_login_ios_sdk.framework"

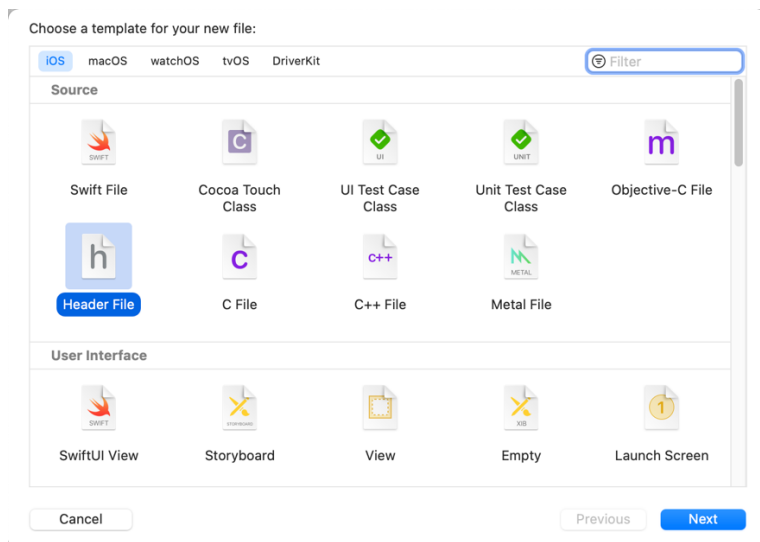


6. build settings 例设置 Other Linker Flags 添加 "-ObjC"



***如果项目是 swift 需要添加如下步骤**

7. 创建 OCBridgeHeader.h 桥接文件



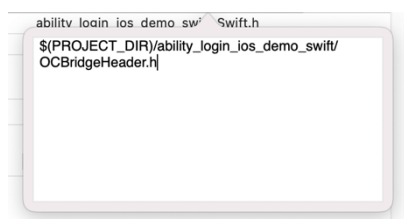
8. 桥接文件中添加如下引用

```
#ifndef OCBridgeHeader_h
#define OCBridgeHeader_h

#import <ability_login_ios_sdk/ability_login_ios_sdk.h>
#import <ability_login_ios_sdk/AbilityLoginManage.h>
#import <ability_login_ios_sdk/ClickLogConfig.h>

#endif /* OCBridgeHeader_h */
```

9. build settings 里设置 Objective-C Bridging Header 设置为 \$(PROJECT_DIR)/ability_login_ios_demo_swift/OCBridgeHeader.h



10. Precompile Bridging Header 设置为 No

Swift Compiler - General

Setting

LXLogSwiftDemo

Precompile Bridging Header

No

2.2 初始化接口

应用调用此方法初始化 sdk , 初始化会校验应用标识是否存在 , 回调成功后进行下一步预取号操作。

```
-(void)initLogin:(NSString *)lxId secret:(NSString *)lxsecret
```

```
isCompletion:(void(^)(NSDictionary * _Nonnull result))completion
```

参数定义

参数名称	含义
lxid	用户申请应用的 id
lxsecret	用户申请应用的 secret (1.0.3 及版本以上)
completion	初始化结果回调

成功响应

参数名称	类型	含义
code	NSString	200
msg	NSString	success
userId	NSString	用户 userId
operatorType	NSString	运营商类型

代码示例：

```
[[AbilityLoginManage instance] initLogin:appId secret:appsecret  
isCompletion:^(NSDictionary * _Nonnull result) {  
    NSString *success = [NSString stringWithFormat:@"%@", [result  
objectForKey:@"code"]];  
    if ([success isEqualToString:@"200"]) {
```

```
        NSLog(@"初始化成功");  
    }  
};
```

2.3 预取号接口

应用调用预取号接口校验应用签名，校验成功后接口会返回运营商类型和状态信息，回调成功后进行调起授权页面操作。

```
-(void)preLogin:(double)timeout isCompletion:(void(^)(NSDictionary *_Nonnull  
result))completion
```

参数定义

参数名称	含义
timeout	超时时间，单位秒
completion	预取号结果回调

成功响应

参数名称	类型	含义
resultCode	NSString	0
resultMsg	NSString	获取 accessCode 成功
operatorType	NSString	运营商类型

代码示例：

```
[[AbilityLoginManage instance] preLogin:8 isCompletion:^(NSDictionary *
```



```
_Nonnull result) {  
  
    if ([result[@"resultCode"] isEqualToString:@"0"]) {  
  
        NSLog(@"初始化成功");  
  
    }  
  
};
```

2.4 调起授权页接口

应用调用该方法会拉起用户授权页面，当用户点击确认授权后接口会返回运营商类型和 token ;用户授权页支持部分页面样式自定义 ,开发者可以通过实例化 ClickLogConfig 类，设置响应的参数调整页面样式。

```
-(void)loginToken:(nonnull UIViewController *)controller clickConfig:(nonnull  
ClickLogConfig *)config getTokenTimeout:(double)timeout  
isCompletion:(void (^)(NSDictionary *_Nonnull result))completion
```

参数定义

参数名称	含义
controller	当前页面
config	ClickLogConfig 实例，自定义授权页
timeout	超时时间，单位秒
completion	授权页结果回调

成功响应

参数名称	类型	含义
resultCode	NSString	0

resultMsg	NSString	登录成功
operatorType	NSString	运营商类型
encrypt	NSString	返回登录加密串

代码示例：

```
[[AbilityLoginManage instance] loginToken:self clickConfig:config  
getTokenTimeout:8 isCompletion:^(NSDictionary * _Nonnull result) {  
  
    NSLog(@"~ %@",result);  
  
    //请求客户服务器取号代码  
  
}];
```

2.5 点击其它登录方式回调

当用户在授权页面点击其他登录方式后，sdk 回调该方法。

-(void)otherLogclick

代码示例：

```
[AbilityLoginManage instance].delegate = self; //添加代理  
abilityManageDelegate  
  
-(void)otherLogclick{  
  
    NSLog(@"点击了其他登录方式");  
  
}
```

2.6 点击关闭授权页按钮回调

添加代理参考 2.5

```
-(void)closeButtonClick
```

2.7 点击授权登录未同意隐私协议回调

```
-(void)uncheckPrivacy;
```

代码示例：

```
-(void)uncheckPrivacy{  
  
    [WHToast showMessage:@"自定义-请同意服务条款"  
  
originY:self.view.frame.size.height-100.0 duration:1 finishHandler:nil];  
  
}
```

2.8 点击隐私协议勾选框回调

```
-(void)privacyAgreementBoxStatus:(BOOL)boxStatus ;
```

2.9 开启二次弹窗后，未勾选隐私协议时点击登录按钮的回调

```
-(void)secondPUPUPAbilityLogclick:(NSDictionary *)privacyAndUrl;
```

注：用户可在此回调中自定义二次弹窗视图

2.10 点击登录按钮回调

```
-(void)abilityLogclick;
```

注：用户可在此回调中自定义加载动画

2.11 授权页关闭完成回调

`-(void)closeAuthViewControllerCompleted;`

2.12 二次弹窗获取 token，仅在开启二次弹窗后生效

`-(void)secondPUPUPGetToken:(BOOL)status;`

2.13 获取手机号能力

详见《一键登录服务端接入文档》

2.14 本机号码校验

`-(void)verificationPhoneNumber:(NSString *)phoneNumber`

`verificatTimeout:(double)timeout isCompletion:(void (^)(NSDictionary *_Nonnull
result))completion;`

参数定义

参数名称	含义
phoneNumber	手机号
timeout	超时时间，单位秒
completion	授权页结果回调

成功响应

参数名称	类型	含义
resultCode	NSString	0
resultMsg	NSString	校验成功
operatorType	NSString	运营商类型

encrypt	NSString	返回校验加密串
---------	----------	---------

代码示例：

```
[[AbilityLoginManage instance] verificationPhoneNumber:_phoneTextStr  
verificatTimeout:8 isCompletion:^(NSDictionary * _Nonnull result) {  
  
    NSLog(@"result%@", result);  
  
    //请求客户服务器号码校验代码  
  
}];
```

2.15判断设备移动数据网络是否开启

```
-(BOOL)checkDeviceCellularDataEnable ;
```

2.16控制台日志输出开关

```
-(void)setPrintLoggerEnable:(BOOL)enable ;
```

2.17获取 sdk 版本号

```
-(NSString *)getVersion ;
```

2.18删除预取号信息

```
-(void)clearLoginCache;
```

2.19获取运营商类型

```
-(NSString *)getOperatorType;
```

2.20 授权页面自定义 UI 参数

ClickLogConfig 属性

#pragma mark **VC属性**

/**授权登录页面是否全屏 默认全屏no*/

@property (nonatomic, assign) BOOL logIsPopup;

#pragma mark **状态栏**

/**状态栏样式*/

@property (nonatomic,assign) UIStatusBarStyle statusBarStyle;

#pragma mark **授权页导航栏**

/**导航栏背景颜色*/

@property (nonatomic,strong) UIColor *navBackColor;

/**导航栏标题文本内容*/

@property (nonatomic,strong) NSString *navText;

/**导航栏标题字体大小*/

@property (nonatomic,assign) int navTextSize;

/**导航栏标题字体是否加粗*/

@property (nonatomic,assign) BOOL navFontBold;

/**导航栏标题字体样式*/

@property (nonatomic,copy) NSString *navTextStyle;

/**导航栏标题字体颜色*/

@property (nonatomic,strong) UIColor *navTextColor;

/**导航栏标题是否隐藏*/

@property (nonatomic,assign) BOOL navIsHide;

/**返回按钮图片*/

@property (nonatomic,strong) UIImage *navReturnImg;

/**返回按钮图片上下位置*/

@property (nonatomic,assign) float navReturnOffsetX; //全屏时：按钮左侧距离屏幕左侧的偏移量
弹框时：按钮左侧距离弹框右侧的偏移量

@property (nonatomic,assign) float navReturnOffsetY;

/**返回按钮图片宽高*/

@property (nonatomic,assign) float navReturnWidth;

@property (nonatomic,assign) float navReturnHeight;

/**返回按钮图片缩放类型*/

@property (nonatomic,assign) UIViewContentMode navReturnContentMode;

/**返回按钮是否隐藏*/

@property (nonatomic,assign) BOOL navReturnHide;

#pragma mark 隐私协议导航栏

/**进入隐私协议导航栏背景颜色，不设置则与授权页一致*/

@property (nonatomic,strong) UIColor *navPruvacyBackColor;

/**进入隐私协议导航栏标题字体大小，不设置则与授权页一致*/

@property (nonatomic,assign) int navPruvacyTextSize;

/**进入隐私协议导航栏标题字体颜色，不设置则与授权页一致*/

@property (nonatomic,strong) UIColor *navPruvacyTextColor;

/**进入隐私协议导航栏返回图片，不设置则与授权页一致*/

@property (nonatomic,strong) UIImage *navPruvacyReturnImg;

#pragma mark LOGO

/**LOGO图片*/

@property (nonatomic,strong) UIImage *logoImg;

/**LOGO的上下左右位置*/

@property (nonatomic,assign) float logoOffsetX;

@property (nonatomic,assign) float logoOffsetY;

```
/**LOGO的宽高*/

@property (nonatomic,assign) float logoWidth;

@property (nonatomic,assign) float logoHeight;

/**LOGO是否隐藏*/

@property (nonatomic,assign) BOOL logoIsHiddle;

/**LOGO图片缩放类型*/

@property (nonatomic,assign) UIViewContentMode logoContentMode;


#pragma mark 手机掩码

/**手机号码字体颜色*/

@property (nonatomic,strong) UIColor *numberColor;

/**手机号码字体大小*/

@property (nonatomic,assign) int numberTextSize;

/**手机号码字体加粗，默认不加粗*/

@property (nonatomic,assign) int numberTextBold;

/**手机号码MidX距离视图MidX轴的偏移*/ //注意不要超出屏幕，超出屏幕移动会错位

@property (nonatomic,assign) float numFieldOffsetX;

/**手机号码Y轴距离顶部偏移*/

@property (nonatomic,assign) float numFieldOffsetY;

/**手机号码文本字体样式，设置后字体加粗属性失效*/

@property (nonatomic,strong) NSString *numberTextStyle;

/**设置手机号码框的高度，暂不支持*/


#pragma mark 认证品牌

/**认证品牌字体大小*/

@property (nonatomic,assign) int brandLabelTextSize;

/**认证品牌字体颜色*/

@property (nonatomic,strong) UIColor *brandLabelTextColor;

/**认证品牌X轴距离屏幕左侧的偏移*/

@property (nonatomic,assign) float brandLabelOffsetX;
```



```
/**认证品牌Y轴距离顶部偏移*/

@property (nonatomic,assign) float brandLabelOffsetY;

/**认证品牌是否加粗*/

@property (nonatomic,assign) BOOL brandFontBold;

/**认证品牌文本字体样式，设置后字体加粗属性失效*/

@property (nonatomic,strong) NSString *brandTextStyle;

/**认证品牌文本自定义数组，固定顺序：移动，联通，电信*/

@property (nonatomic,strong) NSArray *brandTextArray;


#pragma mark 登录按钮

/**登录按钮文本内容*/

@property (nonatomic,strong) NSString *logBtnText;

/**登录按钮字体大小*/

@property (nonatomic,assign) int logBtnTextSize;

/**登录按钮文本颜色*/

@property (nonatomic,strong) UIColor *logBtnTextColor;

/**登录按钮的左右边距 注意:按钮呈现的宽度在竖屏时必须大于屏幕的一半,横屏时必须大于屏幕的三分之一,默认左右是20
*/

@property (nonatomic, assign) CGFloat logBtnOriginL;

@property (nonatomic, assign) CGFloat logBtnOriginR;

/**登录按钮高 注意：必须大于40*/

@property (nonatomic,assign) float logBtnHeight;

/**登录按钮Y轴位置*/

@property (nonatomic,assign) float logBtnOffsetY;

/**登录按钮背景图片数组*/

@property (nonatomic,strong) NSArray *logBtnImgs;

/**登录按钮字体是否加粗*/

@property (nonatomic,assign) BOOL logBtnFontBold;

/**登录按钮文本字体样式，设置后字体加粗属性失效*/
```

```
@property (nonatomic,strong) NSString *logBtnTextStyle;
```

```
#pragma mark 其他登录按钮
```

```
/**其他登录方式是否隐藏*/
```

```
@property (nonatomic,assign) BOOL otherWayLogBtnHidden;
```

```
/**其他登录方式文本内容*/
```

```
@property (nonatomic,strong) NSString *otherWayLogBtnText;
```

```
/**其他登录方式字体颜色*/
```

```
@property (nonatomic,strong) UIColor *otherWayLogBtnTextColor;
```

```
/**其他登录方式文本字体样式，默认 Helvetica*/
```

```
@property (nonatomic,strong) NSString *otherWayLogBtnTextStyle;
```

```
/**其他登录方式字体大小*/
```

```
@property (nonatomic,assign) int otherWayLogBtnTextSize;
```

```
/**其他登录方式上下左右位置*/
```

```
@property (nonatomic,assign) float otherWayLogBtnOffsetX;
```

```
@property (nonatomic,assign) float otherWayLogBtnOffsetY;
```

```
/**其他登录方式宽高*/
```

```
@property (nonatomic,assign) float otherWayLogBtnOffWidth;
```

```
@property (nonatomic,assign) float otherWayLogBtnOffHeight;
```

```
#pragma mark 隐私协议
```

```
/**勾选框未勾选图片*/
```

```
@property (nonatomic,strong) UIImage *uncheckedImg;
```

```
/**勾选框已勾选图片*/
```

```
@property (nonatomic,strong) UIImage *checkedImg;
```

```
/**勾选框是否勾选*/
```

```
@property (nonatomic,assign) BOOL checkedStatus;
```

```
/**勾选框宽高*/  
  
@property (nonatomic,assign) float checkedWH;  
  
/**隐私条款上左右边距*/  
  
@property (nonatomic,assign) float pruvacyOffsetL;  
  
  
@property (nonatomic,assign) float pruvacyOffsetR;  
  
/**隐私条款顶部距离屏幕顶部偏移量，注意：若设置值超出屏幕边界会错位显示*/  
  
@property (nonatomic,assign) float pruvacyOffsetY;  
  
/**隐私条款字体大小*/  
  
@property (nonatomic,assign) int pruvacyTextSize;  
  
/**隐私条款最前面的文字*/  
  
@property (nonatomic,strong) NSString *pruvacyBeforText;  
  
/**第二个协议和第三个协议之间连接文字*/  
  
@property (nonatomic,strong) NSString *pruvacyLinksText;  
  
/**隐私条款最后面的文字*/  
  
@property (nonatomic,strong) NSString *pruvacyAfterText;  
  
/**自定义隐私条款标题和url数组*/  
  
@property (nonatomic,strong) NSArray *pruvacyTitleAndUrlArr;  
  
/**运营商默认隐私协议显示位置，第一个或最后一个*/  
  
@property (nonatomic,assign) DefaultPruvacyPosition defaultPruvacyPosition;  
  
/**隐私条款正常文字颜色*/  
  
@property (nonatomic,strong) UIColor *norTruvacyTextColor;  
  
/**隐私条款协议文字颜色*/  
  
@property (nonatomic,strong) UIColor *pruTruvacyTextColor;  
  
/**运营商隐私条款是否带书名号，默认不带书名号*/  
  
@property (nonatomic,assign) BOOL pruvacyBookMark;  
  
/**运营商隐私条款行距*/  
  
@property (nonatomic,assign) float pruvacyLineSpacing;  
  
/**运营商隐私条款文本对齐方式*/  
  
@property (nonatomic,assign) NSTextAlignment pruvacyAlignment;
```

/**运营商隐私条款文本字体样式，设置后字体加粗属性失效*/

@property (nonatomic,strong) NSString *pruvacyTextStyle;

/**运营商隐私条款文本加粗*/

@property (nonatomic,assign) BOOL pruvacyFontBold;

#pragma mark 授权页面

/**授权页弹出和消失方向，仅支持全屏授权页*/

@property (nonatomic,assign) PresentDismissDirection presentDismissDirection;

/**背景颜色*/

@property (nonatomic,strong) UIColor *backgroundColor;

/**背景图片*/

@property (nonatomic,strong) UIImage *backgroundImage;

#pragma mark 弹窗授权页

/**弹窗的宽高比例，底部弹窗时popupScaleW设置无效*/

@property (nonatomic,assign) float popupScaleW;

@property (nonatomic,assign) float popupScaleH;

/**弹窗的圆角，底部弹窗时只对弹窗左上角和右上角生效*/

@property (nonatomic,assign) float popupCornerRadius;

/**是否底部弹窗，true 底部弹窗，false 居中弹窗*/

@property (nonatomic,assign) BOOL isBottom;

/**弹窗授权页背景透明度，仅支持居中弹窗*/

@property (nonatomic,assign) float popupAlpha;

/**弹框授权页上下左右位置，暂不支持*/

#pragma mark 全屏页二次弹窗，仅在全屏时可用

/**全屏页面是否开启自定义二次弹窗*/

@property (nonatomic,assign) BOOL secondPOPUP;

```
#pragma mark 自定义控件
```

```
typedef void(^myCustomUIHandler)(UIView *customAreaView);
```

```
/**添加自定义控件的，三方入口*/
```

```
@property (nullable, nonatomic, copy) myCustomUIHandler customUIHandler;
```