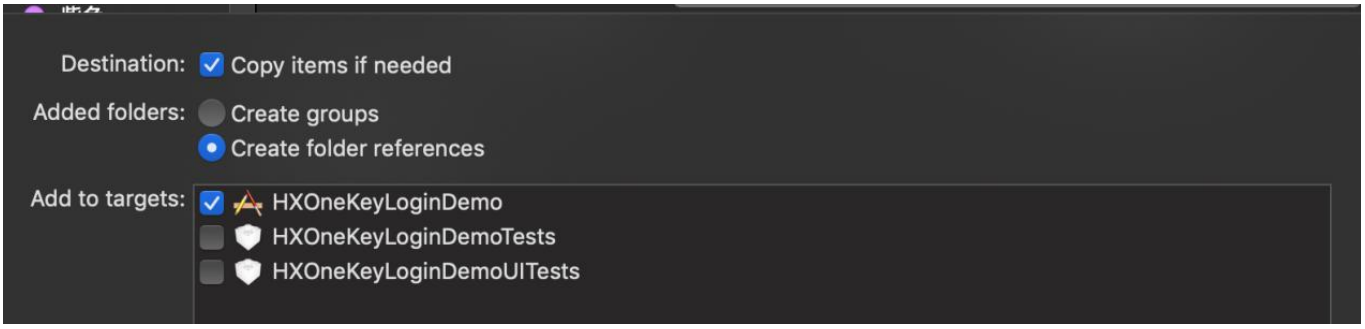
iOS 集成文档

1.快速集成

1.1 目录解构

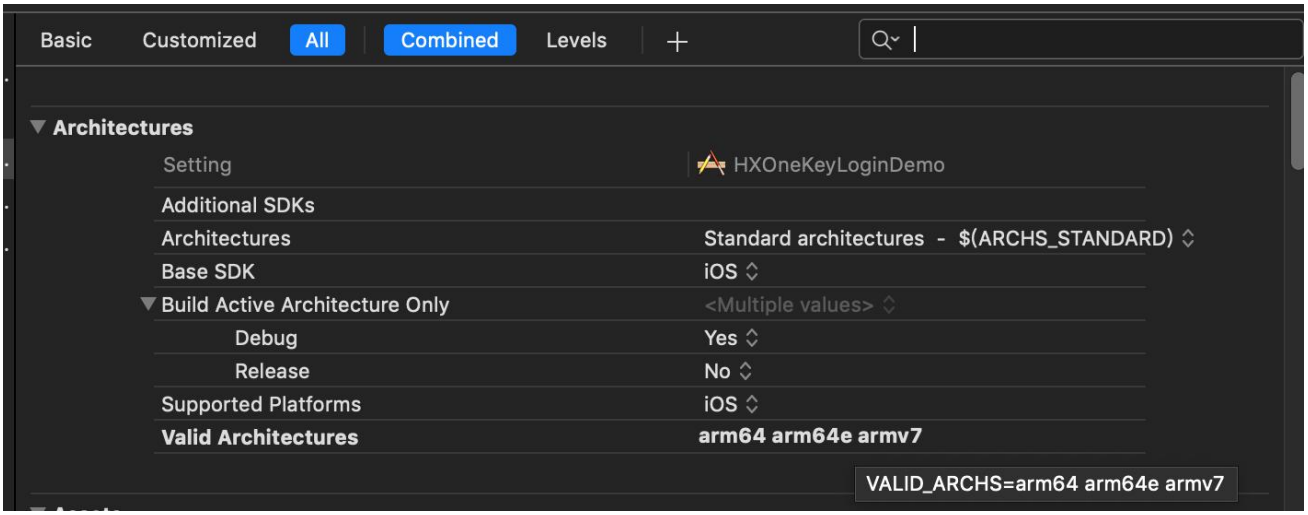
----HXOneKeyLogin-iOS.zip
----|----HXOneKeyLoginDemo(demo 示例)
----|----HXLogin-V1.0.0（Framework）
----|----|----CustomUI（授权页面布局文件需导入）
----|----|----I----XYOKLAuthVC_LF.xib（横屏全屏沉浸式）
----|----|----I----XYOKLAuthVC_LW.xib（横屏弹框式）
----|----|----I----XYOKLAuthVC_PF.xib（竖屏全屏沉浸式）
----|----|----I----XYOKLAuthVC_PW.xib（竖屏弹框式）
----|----|----I----XYOKLWebNavController.h（协议的导航控制器）
----|----|----I----XYOKLWebNavController.m
----|----|----I----XYOKLWebVC.h（协议的 webview）
----|----|----I----XYOKLWebVC.m
----|----|----HXLogin.framework（framework）
----|----|----EAccountOpenPageResource.bundle
----|----|----sdk_oauth.bundle
----|----|----TYRZResource.bundle
----|----|----XYOKLResource.bundle

1.2 将 HXLogin-V1.0.0 导入到项目，选择 Copy items if needed，如图所示

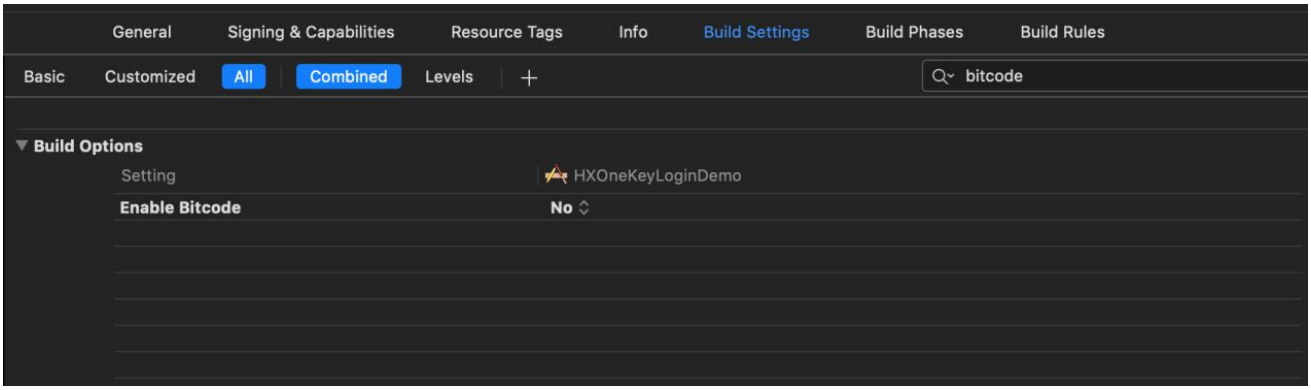


1.3 info.plist 文件中添加 App Transport Security Settings 并设置 Allow Arbitrary Loads 为 YES

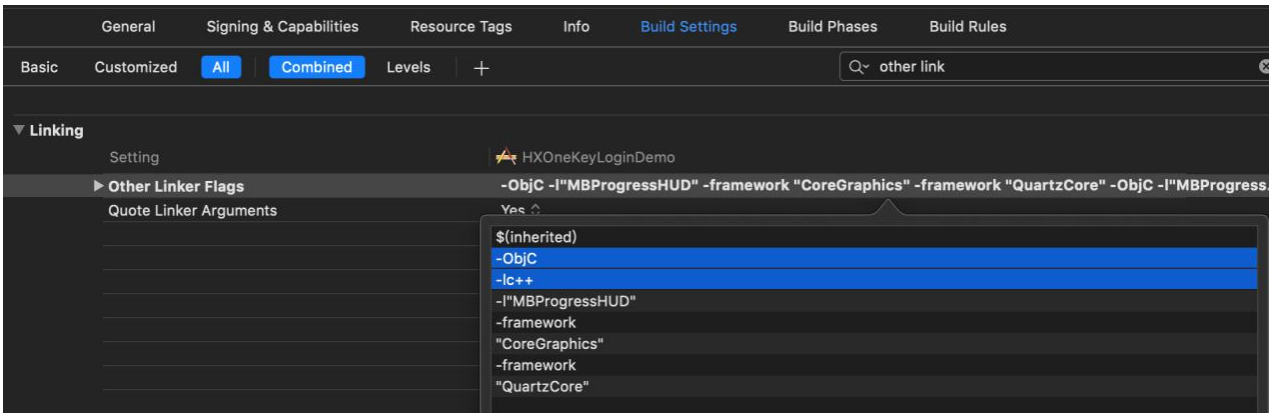
1.4 修改所支持的架构。目前支持 arm64、armv7、arm64e 架构



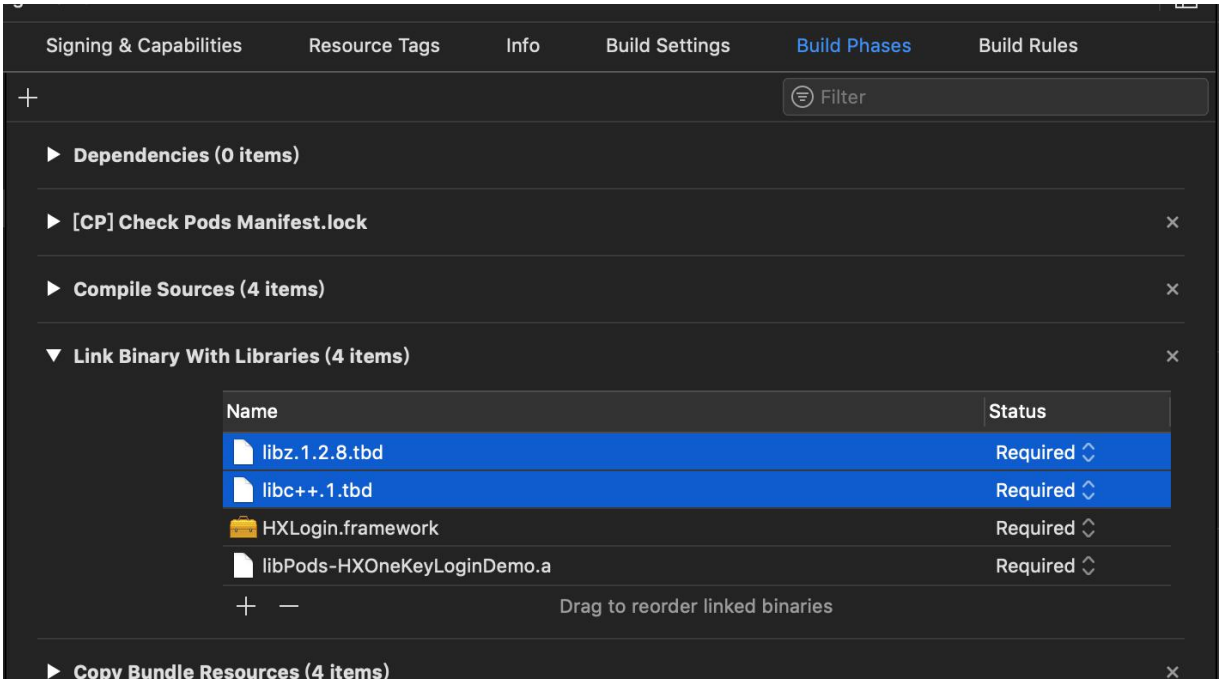
1.5 设置 Enable Bitcode 为 No。



1.6 添加 Xcode 链接器参数：-ObjC、-lc++



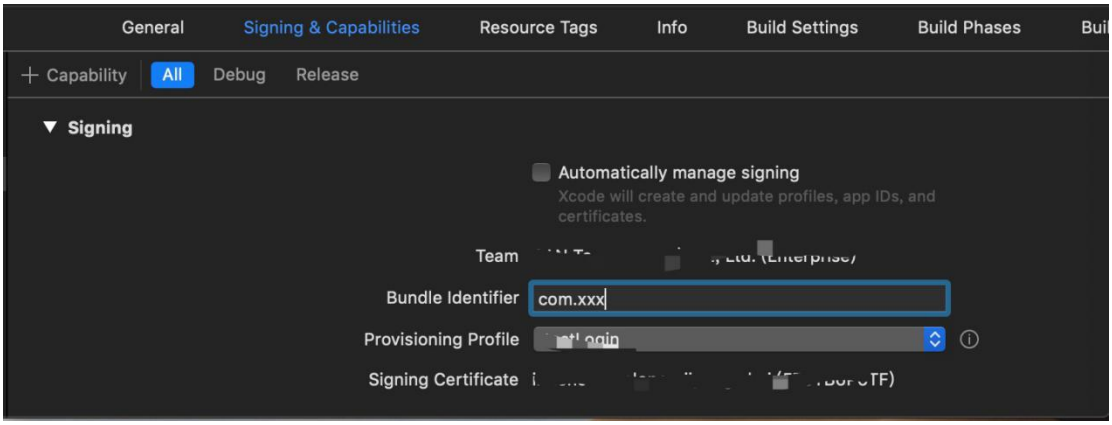
1.7 添加依赖库，在项目设置 target -> 选项卡 Build Phases -> Link Binary with Libraries 下添加 libc++.1.tbd 和 libz.1.2.8.tbd



1.8 替换 MyPrefixHeader.pch 中的初始化宏定义参数

```
#define HXLogionDemo_AppKey @"应用申请获取的 appKey"
#define HXLogionDemo_AppSecret @"应用申请获取的 AppSercret"
#define HXLogionDemo_ClientId @"应用申请获取的用户 id"
```

1.9 填写平台注册提供的 bundle id



2.SDK 方法说明

2.1 初始化 SDK

```
/**
 初始化 SDK
 @param appKey 平台生产的 appKey
 @param appSecret 平台生产的 appSecret
 @param clientId 平台生产的商户号
 */
+ (void)hxAppKey:(NSString *)appKey
       appSecret:(NSString *)appSecret
        clientId:(NSString *)clientId;
```

代码示例

```
//导入 sdk 头文件
#import <HXLogin/HXLogin.h>
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.
    ....
[HXOneKeyLogin hxAppKey:HXLogionDemo_AppKey
appSecret:HXLogionDemo_AppSecret  clientId:HXLogionDemo_ClientId];

    ....
    }
```

2.2 SDK 登录方法

2.2.1 SDK 登录方式一

2.2.1.1 预取号

```
/**
 登录方式一 （第一步：预取号）
 @param resultBlock 预取号结果
 */
+ (void)getPreLoginPhoneNumber:(HXOKLoginBlock)resultBlock;
```

方法作用

经历校验网络和配置、运营商预取号两个过程，为快速拉起授权页做准备。

此方法为异步执行。

请在此方法调用成功之后，再调用拉起授权页方法进行用户授权操作。

参数说明

```
/**
 SDK 方法回调
 @param success 登录是否成功
 @param error 登录失败错误回调
 @param result 登录成功回调
 */
typedef void(^HXOKLoginBlock)(BOOL success,id error,id result);
```

参数名称	参数类型	参数说明
success	BOOL	YES：预取号成功；NO：预取号失败
error	NSDictionary	success 为 NO 时有值，否则为 nil
result	NSDictionary	都为 nil

示例

可参考 demo 中示例代码

2.2.1.2 拉起授权页

```
/**
 登录方式一 （第二步：拉取授权页）
```

```
 @param clickResult 点击事件回调,返回点击事件的 tag
 @param loginResult 登录结果回调
 */
+ (void)openAuthLoginClickResult:(HXOKLoginClickBlock)clickResult
                     loginResult:(HXOKLoginBlock)loginResult;
```

方法作用

拉起用户授权页，经历运营商登录、登录成功或失败回调两个过程。

在预取号方法成功后，再调用此方法拉起授权页。

使用登录成功回调的 oclToken 与服务端接口交互，获得明文手机号。参考服务端 REST API 对接指南，一键登录获取手机号 API 部分。

参数说明

```
/**
自定义按钮的点击回调
 @param senderTag 点击按钮的 tag 值
 */
typedef void(^HXOKLoginClickBlock)(NSString *senderTag);
```

注意:

1、自定义按钮的 tag 值范围是 9000~9100;

2、tag 为 9200，代表未勾选协议时，点击一键登录按钮的回调;

3、tag 为 9300，代表已勾选协议时，点击一键登录按钮的回调。

示例代码

```
    [self showHUDProgressIsOnKeyWindow:YES];
    __weak typeof(self)weakSelf = self;
    //登录方式一 （第一步：预取号）
    [HXOneKeyLogin getPreLoginPhoneNumber:^(BOOL success, id error, id result) {
        if (success) {
            //登录方式一 （第二步：拉取授权页）
            [HXOneKeyLogin openAuthLoginClickResult:^(NSString *senderTag) {
                [weakSelf handleClickResult:senderTag];
            } loginResult:^(BOOL success, id error, id result) {
                [weakSelf handleLoginResult:success error:error result:result];
            }];
        } else {
            [weakSelf handleLoginResult:success error:error result:result];
        }
    }];
#pragma mark - SDK 点击事件处理
```

```
-(void)handleClickResult:(NSString *)senderTag
{
    dispatch_async(dispatch_get_main_queue(), ^{
        [self hideHUDProgress];
        if ([senderTag isEqualToString:@"9200"]) {
            [self showMessageHUDWith:@"请同意服务条款" autoHide:YES];
        } else if ([senderTag isEqualToString:@"9300"]) {
            [self showMessageHUDWith:@"正在登录中..." autoHide:NO];
        } else {
            [self showMessageHUDWith:[NSString stringWithFormat:@"点击事件回调 tag
值：%@",senderTag] autoHide:YES];
        }
    });
}
#pragma mark - SDK 结果展示
-(void)handleLoginResult:(BOOL) success error:(id)error result:(id)result
{
    dispatch_async(dispatch_get_main_queue(), ^{
        //UI 处理
        [self hideHUDProgress];
        NSString *messageString = @"";
        if (success) {
            //登录成功处理
            NSString *token =  result[@"oclToken"]; //平台授权 token
            NSString *operatorType =  result[@"operatorType"]; //运营商类别 CU 联通 CT
电信 CM 移动
            NSLog(@"平台授权 oclToken = %@\n 运营商类别 operatorType = %@\n（CU 联通 CT 电信
CM 移动）",token,operatorType);
            messageString = [NSString stringWithFormat:@"平台授权 oclToken = %@\n 运营商
类别 operatorType = %@\n（CU 联通 CT 电信 CM 移动）",token,operatorType];
        } else {
            //登录失败处理
            NSString *errorCode =  error[@"errorCode"]; //错误码
            NSString *errorMsg =  error[@"errorMsg"]; //错误描述
            NSString *errorDetailMsg =  error[@"errorDetailMsg"]; //详细描述
            NSString *traceNo =  error[@"traceNo"]; //追踪号
            NSLog(@"错误码 errorCode = %@\n 错误描述 errorMsg = %@\n 详细描述 errorDetailMsg
= %@\n 追踪号 traceNo = %@",errorCode,errorMsg,errorDetailMsg,traceNo);
            messageString = [NSString stringWithFormat:@"错误码 errorCode = %@\n 错误描
述 errorMsg = %@\n 详细描述 errorDetailMsg = %@\n 追踪号 traceNo
= %@",errorCode,errorMsg,errorDetailMsg,traceNo];
        }
        [self showAlertWithMessage:messageString isSuccess:success];
    });
}
```

2.2.2 SDK 登录方式二

```
/**
 登录方式二
 @param clickResult 点击事件回调,返回点击事件的 tag
 @param loginResult 登录结果回调
 */
+ (void)startLoginClickResult:(HXOKLoginClickBlock)clickResult
                  loginResult:(HXOKLoginBlock)loginResult;
```

方法作用

> 经历校验网络和配置、运营商预取号、运营商登录、登录成功或失败回调四个过程，是方式一的步骤合并。
>
> 使用登录成功回调的 oclToken 与服务端接口交互，获得明文手机号。参考服务端 REST API 对接指南，一键登录获取手机号 API 部分。

参数说明可参考登录方式一。

注意:

1、自定义按钮的 tag 值范围是 9000~9100;

2、tag 为 9200，代表未勾选协议时，点击一键登录按钮的回调;

3、tag 为 9300，代表已勾选协议时，点击一键登录按钮的回调。

示例代码

```
    [HXOneKeyLogin startLoginClickResult:^(NSString *senderTag) {
        [weakSelf handleClickResult:senderTag];
    } loginResult:^(BOOL success, id error, id result) {

          [weakSelf handleLoginResult:success error:error result:result];

    }];
#pragma mark - SDK 点击事件处理
-(void)handleClickResult:(NSString *)senderTag
{

     dispatch_async(dispatch_get_main_queue(), ^{
```

```
        [self hideHUDProgress];

        if ([senderTag isEqualToString:@"9200"]) {
            [self showMessageHUDWith:@"请同意服务条款" autoHide:YES];
        } else if ([senderTag isEqualToString:@"9300"]) {
            [self showMessageHUDWith:@"正在登录中..." autoHide:NO];
        } else {
            [self showMessageHUDWith:[NSString stringWithFormat:@"点击事件回调 tag
值：%@",senderTag] autoHide:YES];
        }
    });
}
#pragma mark - SDK 结果展示
-(void)handleLoginResult:(BOOL) success error:(id)error result:(id)result
{
    dispatch_async(dispatch_get_main_queue(), ^{
        //UI 处理
        [self hideHUDProgress];
        NSString *messageString = @"";
        if (success) {
            //登录成功处理

            NSString *token =  result[@"oclToken"]; //平台授权 token

            NSString *operatorType =  result[@"operatorType"]; //运营商类别 CU 联通 CT
电信 CM 移动
            NSLog(@"平台授权 oclToken = %@\n 运营商类别 operatorType = %@\n（CU 联通 CT 电信
CM 移动）",token,operatorType);
            messageString = [NSString stringWithFormat:@"平台授权 oclToken = %@\n 运营商
类别 operatorType = %@\n（CU 联通 CT 电信 CM 移动）",token,operatorType];
        } else {
            //登录失败处理
            NSString *errorCode =  error[@"errorCode"]; //错误码
            NSString *errorMsg =  error[@"errorMsg"]; //错误描述
            NSString *errorDetailMsg =  error[@"errorDetailMsg"]; //详细描述
            NSString *traceNo =  error[@"traceNo"]; //追踪号
            NSLog(@"错误码 errorCode = %@\n 错误描述 errorMsg = %@\n 详细描述 errorDetailMsg
= %@\n 追踪号 traceNo = %@",errorCode,errorMsg,errorDetailMsg,traceNo);
            messageString = [NSString stringWithFormat:@"错误码 errorCode = %@\n 错误描
述 errorMsg = %@\n 详细描述 errorDetailMsg = %@\n 追踪号 traceNo
= %@",errorCode,errorMsg,errorDetailMsg,traceNo];
        }
        [self showAlertWithMessage:messageString isSuccess:success];
    });
}
```

2.3 本机号码认证方法

2.3.1 本机号码认证方法一

2.3.1.1 预取号

```
/**
 号码认证方式一 （第一步： 预取号）
 @param resultBlock 预取号回调结果
 */
+ (void)getPreMobileAuthPhoneNumber:(HXOKLoginBlock)resultBlock;
```

示例代码

```
[HXOneKeyLogin getPreLoginPhoneNumber:^(BOOL success, id error, id result) {

    }];
```

2.3.1.2 号码认证

```
/**
 号码认证方式一 （第二步：获取号码认证 token）
 @param resultBlock 获取号码认证回调结果
 */
+ (void)mobileAuth:(HXOKLoginBlock)resultBlock;
```

方法作用

> 必须在号码认证预取号之后调用，进行号码认证。
>
> 获得 oclToken 和运营商类型，使用 oclToken 与服务端接口交互，验证号码是否一致。参考服务端 REST API 对接指南，**本机号码认证 API** 部分。

示例代码

```
    [self showHUDProgressIsOnKeyWindow:YES];
    __weak typeof(self)weakSelf = self;
    [HXOneKeyLogin getPreMobileAuthPhoneNumber:^(BOOL success, id error, id result) {
        if (success) {

            [HXOneKeyLogin mobileAuth:^(BOOL success, id error, id result) {
```

```
                dispatch_async(dispatch_get_main_queue(), ^{
                    //UI 处理
                    [weakSelf hideHUDProgress];
                    if (success) {
                        //成功处理
                        NSString *token =  result[@"oclToken"]; //平台授权 token
                        NSString *operatorType =  result[@"operatorType"]; //运营商类别 CU 联通 CT 电信 CM 移动
                        NSLog(@"成功操作 token = %@, operatorType = %@",token,operatorType);
                        NSString *reasonStr = [NSString stringWithFormat:@"Token: %@\n运营商类别: %@ \n(CU 联通 CT 电信 CM 移动)",token,operatorType];
                    } else {
                        //失败处理
                        NSString *errorCode =  error[@"errorCode"]; //错误码
                        NSString *errorMsg =  error[@"errorMsg"]; //错误描述
                        NSString *errorDetailMsg =  error[@"errorDetailMsg"]; //详细描述
                        NSString *traceNo =  error[@"traceNo"]; //追踪号
                        NSLog(@"失败操作 errorCode = %@, errorMsg = %@, errorDetailMsg = %@, traceNo = %@",errorCode,errorMsg,errorDetailMsg,traceNo);
                    }
                });
            }];
        } else {
            dispatch_async(dispatch_get_main_queue(), ^{
                [weakSelf hideHUDProgress];
                //失败处理
                NSString *errorCode =  error[@"errorCode"]; //错误码
                NSString *errorMsg =  error[@"errorMsg"]; //错误描述
                NSString *errorDetailMsg =  error[@"errorDetailMsg"]; //详细描述
                NSString *traceNo =  error[@"traceNo"]; //追踪号
                NSLog(@"失败操作 errorCode = %@, errorMsg = %@, errorDetailMsg = %@, traceNo = %@",errorCode,errorMsg,errorDetailMsg,traceNo);
            });
        }
    }];
```

2.3.2 本机号码认证方式二

```
/**
 号码认证方法二
 @param resultBlock 获取号码认证回调结果

 */
+ (void)startCertificationWithResultBlock:(HXOKLoginBlock)resultBlock;
```

方法作用

此方法为号码认证方式一的步骤合并。

获得 oclToken 和运营商类型，使用 oclToken 与服务端接口交互，验证号码是否一致。参考服务端 REST API 对接指南，**本机号码认证 API** 部分。

示例代码

```
    [self showHUDProgressIsOnKeyWindow:YES];
    __weak typeof(self)weakSelf = self;
    [HXOneKeyLogin startCertificationWithResultBlock:^(BOOL success, id error, id result) {
        dispatch_async(dispatch_get_main_queue(), ^{
            //UI 处理
            [weakSelf hideHUDProgress];
            if (success) {
                //成功处理
                NSString *token =  result[@"oclToken"]; //平台授权 token
                NSString *operatorType =  result[@"operatorType"]; //运营商类别 CU 联通 CT 电信 CM 移动
                NSLog(@"成功操作 token = %@, operatorType = %@",token,operatorType);
                NSString *reasonStr = [NSString stringWithFormat:@"Token: %@\n 运营商类别: %@ \n(CU 联通 CT 电信 CM 移动)",token,operatorType];
            } else {
                //失败处理
                NSString *errorCode =  error[@"errorCode"]; //错误码
                NSString *errorMsg =  error[@"errorMsg"]; //错误描述
                NSString *errorDetailMsg =  error[@"errorDetailMsg"]; //详细描述
                NSString *traceNo =  error[@"traceNo"]; //追踪号
                NSLog(@"失败操作 errorCode = %@, errorMsg = %@, errorDetailMsg = %@, traceNo = %@",errorCode,errorMsg,errorDetailMsg,traceNo);
            }
        });
    }];
```

2.4 自定义 UI 配置

```
/**
 自定义登录页面 UI 配置
 @param config 自定义 UI 配置
 */
+ (void)customUIWithHXOKConfig:(HXOneKeyUIConfig*)config;
```

特别说明

该方法主要设置页面显示方式和协议配置。

该方法请在”SDK 登录方法”之前调用。

HXOneKeyUIConfig 对象属性请参考下文

3.UI 配置

3.1 配置方式

- api（主要设置页面弹出方式和协议配置等参数）
- xib（修改界面控件布局及 tag 值）
- 代码（修改跳转协议的界面）

- 3.2 api 通过 HXOneKeyUIConfig 对象属性设置，并调用 customUIWithHXOKConfig 方法设置 UI 对象。

HXOneKeyUIConfig 属性列表（见 sdk 中的 HXOneKeyUIConfig .h 文件），详情请参考头文件，注意标红的地方

3.3 xib

3.3.1 修改授权界面 UI 的 xib 文件，分别为 XYOKLAuthVC_LF.xib (横屏沉浸式)、XYOKLAuthVC_LW.xib（横屏弹框式）、XYOKLAuthVC_PF.xib（竖屏沉浸式）、XYOKLAuthVC_PW.xib（竖屏弹框式），可根据需求对其修改。

3.3.2 编译工程，进入 Products 下的 .app 文件，显示包内容，将里面的 XYOKLAuthVC_LF.nib（横屏沉浸式）、XYOKLAuthVC_LW.nib（横屏弹框式）、XYOKLAuthVC_PF.nib（竖屏沉浸式）、XYOKLAuthVC_PW.nib（竖屏弹框式）文件替换到，工程的 XYOKLResource.bundle 里

注意：

1、修改 xib 文件里的控件时，请务必检查设置的 tag 值，控件 tag 值参考下文，建议修改 xib 控件里的内容，避免删除后再重新添加时，遗漏了 tag 值的设置；

2、自定义按钮事件的回调请在 4.2.1.2 拉起授权页 和 4.2.3 SDK 登录方式二 的 HXOKLoginClickBlock 中处理，自定义按钮的 tag 值范围 9000~9100；

3、如要修改使用 xib 中使用的图片素材，请务必将图片素材放在 XYOKLResource.bundle 里，通过 api 使用的图片资源，放在 app 自己的工程即可。

3.3.3 xib 文件里的控件 tag 值

登录界面控件 tag 值规定

- 1.左上角导航返回按钮 ———— 21001
- 2.脱敏号码标签 ———— 21002
- 3.logo 图片 ———— 32001
- 4.服务提供标签 ———— 21004
- 5.登录按钮 ———— 21005
- 6.协议勾选框按钮 ———— 21007
- 7.服务与隐私协议标签 ———— 21008
- 8.用户自定义按钮 ———— 9000~9100

登录按钮的点击回调处理

- 1.未勾选协议时，点击登录按钮，回调的 tag 值 ———— 9200

- 2.已勾选协议时，点击登录按钮，回调的 tag 值 ———— 9300

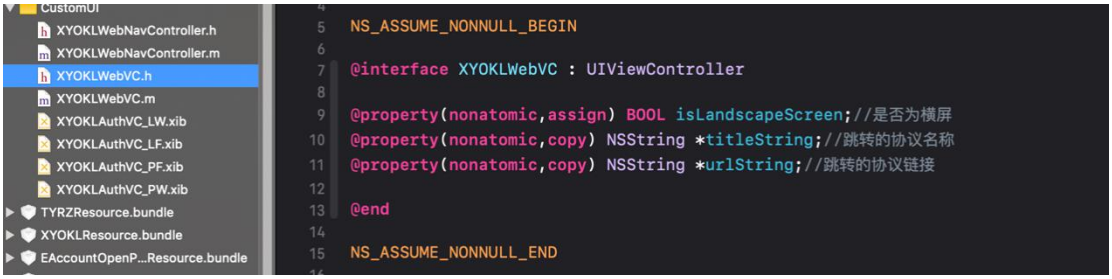
3.4 代码

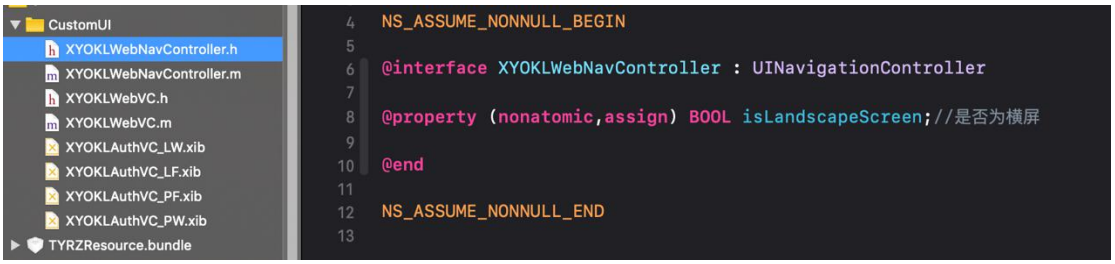
通过修改源码的方式，自定义跳转协议的界面

XYOKLWebNavController

XYOKLWebVC

注意：XYOKLWebNavController.h 和 XYOKLWebVC.h 中的属性请不要修改名字





1.响应码

响应码	描述	描述详情
GET_CONFIGURATION_ERROR	获取 SDK 配置信息失败	对接可能存在错误，请检查配置或联系我们支持
APP_CONFIG_NOT_EXIST	亲，应用配置信息不存在，请确认	对接可能存在错误，请检查配置或联系我们支持
GET_ACCESSCODE_ERROR	运营商预取号失败	用户网络不稳定，可切换纯

		4G 操作
LOGIN_FAILURE	预取号失败	预下单接口失败，联系我们给予支持
CANCEL	登录取消	预取号成功后，用户点击返回
OPERATOR_INIT_FAILED	请求参数有误	对接可能存在错误，请检查配置或联系我们支持
INTERNET_UNAVAULABLE	运营商网络异常	未检测到用户数据流量网络，需要用户检查手机网络情况
PARAM_APP_ID_VALID_NOT_PASS	请求参数 appKey 不能为空	对接可能存在错误，请检查配置或联系我们支持
PARAM_APP_SECRET_VALID_NOT_PASS	请求参数 appSecret 不能为空	对接可能存在错误，请检查配置或联系我们支持
PARAM_MERCHANT_NO_VALID_NOT_PASS	请求参数 merchantNo 不能为空	对接可能存在错误，请检查配置或联系我们支持
PARAM_CALL_BACK_VALID_NOT_PASS	回调函数 callBack 不能为空	对接可能存在错误，请检查配置或联系我们支持
CHECK_PERMISSION_IS_OPEN	请使用运营商网络	手机上网权限没有给 APP
SERVICE_UNAVAILABLE	亲，服务不可用，请稍后重试!	系统内部异常
UNKOWN_CARRIER	未知运营商	未识别的 sim 卡
GET_ACCESSTOKEN_ERROR	运营商登录失败	获取运营商 token 失败，可能是网络不稳定或者未开启数量流量网络

BUNDLE_ID_ERROR	bundle Id 不匹配	对接可能存在错误，请检查配置或联系我们支持
PARAM_DECRYPT_ERROR	请求报文解析失败	对接可能存在错误，请检查配置或联系我们支持
OPERATOR_RESTRICT_REQUESTS	请求次数超限	余额或权限不足、用户频繁操作
OPERATOR_NUMBER_FAILED	运营商取号失败	用户网络不稳定，可切换纯 4G 操作
SIMULATOR_NOT_SUPPORTED	请使用真机运行	一键登录和号码认证不支持模拟器
BUNDLE_RESOURCES_ERROR	请添加 XYOKLResource.bundle 资源文件	请添加 XYOKLResource.bundle 资源文件
BUNDLE_RESOURCES_NIB_ERROR	请在 XYOKLResource.bundle 中添加 nib 资源文件	请在 XYOKLResource.bundle 中添加 nib 资源文件
N0_GETPRENUMBER_ERROR	请先调用相应业务的预取号方法	请先调用相应业务的预取号方法
OTHER_ERROR	其它异常	其它异常、SDK 内部异常