**手机盾SDK集成文档**

1.简介 3

1.1功能介绍 3

1.2兼容性 3

2.集成指南 4

2.1集成SDK 4

2.2集成后系统配置 7

3.接口功能代码 9

3.1初始化 9

3.2接口说明 9

个人申请证书接口 9

企业申请证书接口 11

虚拟申请证书接口 12

原文签名接口 13

原文验签接口 14

原文加密接口 15

原文解密接口 16

重置PIN码接口 17

更新证书接口 17

查询证书接口 18

变更证书状态接口 18

延期证书接口 19

重置缓存密码时间接口 19

证书签章接口 20

证书登录接口 21

获取二维码解绑接口 22

扫码解绑接口 22

清除本地证书 23

签名图片数据存储 23

手写面板签名图片数据存储 24

获取签名图片数据 24

手写签名面板获取图片 25

判断证书是否存在 25

获取本地证书信息 25

获取设备标识 26

4.附录说明： 26

# 1.简介

## 1.1功能介绍

手机盾SDK iOS版是一种数字证书管理的开发包，此版SDK包括身份的认证，证书下载，数字签名等功能。基于该方案，开发者可以轻松的构建该SDK的应用。在使用该SDK之前，我们首先介绍SDK本身的基本功能，以便集成商能够更方便的使用。

此版SDK包含的功能如下：

实名认证：该SDK提供了实名认证的功能，根据集成方的需求可进行人脸识别、银行认证和企业认证这三个认证方式。

证书下载：业务方用过传入用户的信息即可进行证书下载。

签名功能：可进行数字签名，验签等功能。

证书管理：包括证书的更新、延期、变更等操作。

## 1.2兼容性

系统：支持iOS9.0以上系统。需要开发者通过Deployment Target 来保证支持系统的检测。

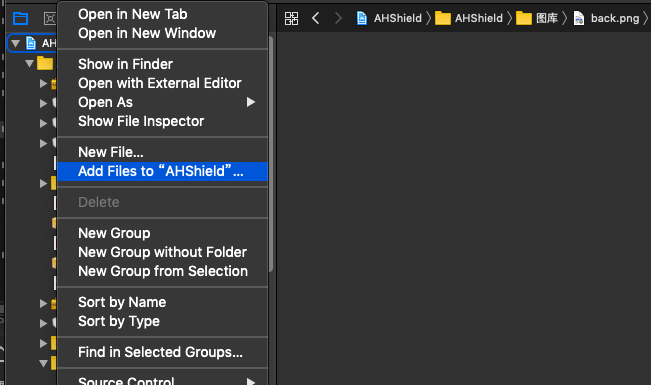
机型：iPhone

网络：支持Wi-Fi及移动网络

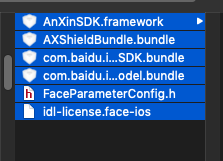
# 2.集成指南

## 2.1集成SDK

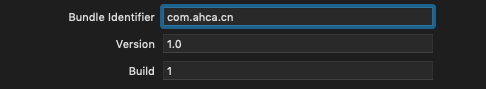
打开或者新建一个项目，右键点击项目添加菜单，在菜单中选择『Add Files to“此处是你的项目名字”…… 』,如下图所示：



在添加文件弹出框里选择SDK内容添加进来。如下图：



有身份认证功能的需提前向我们提供Bundle Identifier( 身份认证时必须要保证Bundle Identifier和授权文件idl\_license.face-ios一致，否则无法进行人脸识别身份认证)

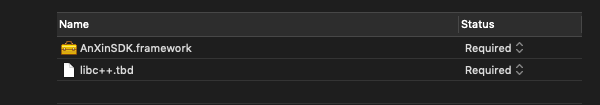


下图是FaceParameterConfig.h文件信息：



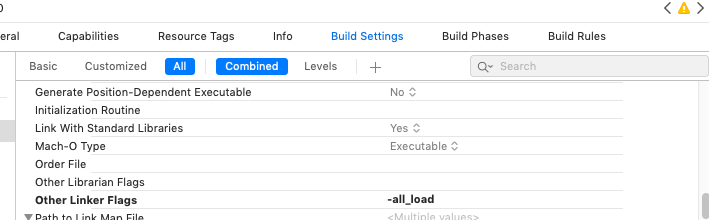
说明：人脸识别时需要idl\_license.face-ios文件进行验证，所以需要该文件中的三个参数。

SDK导入完成后选择链接C++标准库 libc++.tbd，如下图：

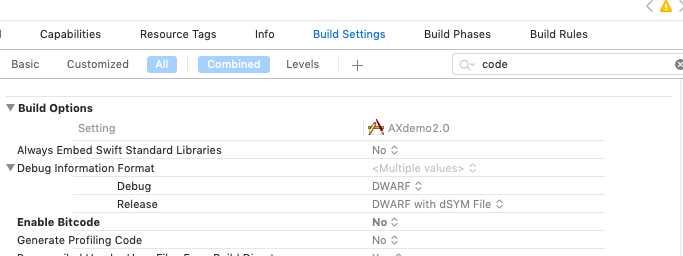


## 2.2集成后系统配置

1）请在Build Setting >Linking > Other Linker Flags 上面加入 –all\_load 选项，如下图所示：

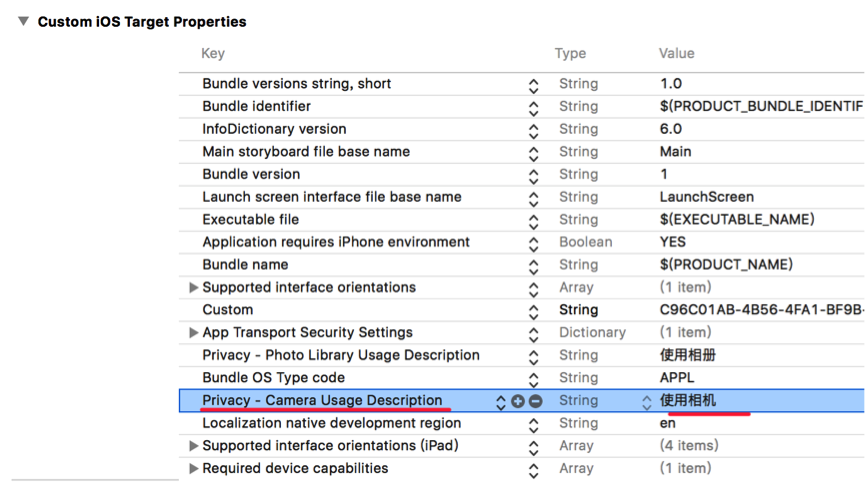


2）请设置Build Setting >Build Options> Enable Bitcode为NO，如下图所示：



3）需要使用相机权限：编辑Info.plist文件，添加

Privacy- Camera Usage Description 的Key值，Value为使用相机时候的提示语，可以填写：『使用相机』。



注：需开启Face ID权限



# 3.接口功能代码

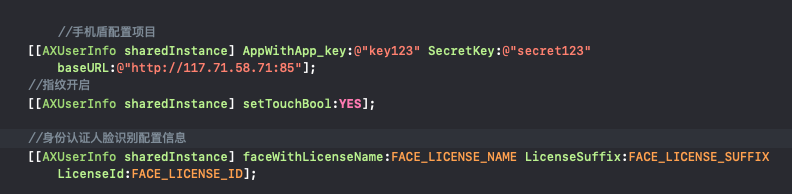
## 3.1初始化

导入头文件

#import <AnXinSDK/AnXinSDK.h>

#import "FaceParameterConfig.h"

在appdelegate中初始化以下代码：



说明：指纹默认是关闭状态，根据集成方的选择可自行配置；若集成方没有身份认证人脸识别功能，则不需要进行“身份认证人脸识别配置信息”初始化

（**注意：URL地址格式，格式不正确会导致接口调用报错）**

## 3.2接口说明

## 个人申请证书接口

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现个人证书的在线申请 | | | |
| 个人证书申请函数原型 |  | | -(**void**)ApplyWithPersonUserID:(NSString \*)userID                        User\_name:(NSString \*)user\_name                         card\_num:(NSString \*)card\_num                        phone\_num:(NSString \*)phone\_num                        card\_type:( CardType)card\_type                       user\_city:(NSString \*)user\_city                          user\_email:(NSString \*)user\_email                        cert\_ext2:(NSString \*)cert\_ext2                        cert\_ext3:(NSString \*)cert\_ext3                        cert\_ext4:(NSString \*)cert\_ext4                         delegate:(**id**<CertDelegate>)delegate; | | | |
| 个人证书申请实现方法 |  | | [[AXUserInfo sharedInstance] ApplyWithPersonUserID:@“111” User\_name:@"张三" card\_num:@"342401199001011234" phone\_num:@"18012345678" card\_type:IDCard user\_city:@"" user\_email:@"" cert\_ext2:nil cert\_ext3:nil cert\_ext4:nil delegate:self];  (说明: 需使用代理方法CertDelegate  获取回调证书信息，代理方法-(void)CertDictionary:(NSDictionary \*)dictionary | | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 | |
| userID | NSString | | Y | | 用户唯一标识 |
| user\_name | NSString | | Y | 用户名（申请人姓名） | |
| card\_num | NSString | | Y | 证件号 | |
| phone\_num | NSString | | N | 手机号码 | |
| card\_type | CardType | | Y | /\*\* 身份证 \*/      IDCard,      /\*\* 临时身份证 \*/      InterimIDCard,       /\*\* 户口本 \*/      HouseholdRegister,       /\*\* 护照 \*/      Passport,       /\*\* 军人身份证 \*/      SoldierIDCard,       /\*\* 武警身份证 \*/      ArmedPoliceIDCard, | |
| user\_city | NSSting | | N | 城市 | |
| user\_email | NSString | | N | 邮箱 | |
| cert\_ext2 | NSString | | N | 扩展项2（证书的用户额外信息项，没有则不传） | |
| cert\_ext3 | NSString | | N | 扩展项3（证书的用户额外信息项，没有则不传） | |
| cert\_ext4 | NSString | | N | 扩展项4（证书的用户额外信息项，没有则不传） | |
| Delegate | Delegate | | Y | 回调方法 | |

## 企业申请证书接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现企业证书的在线申请 | | |
| 企业证书申请函数原型 |  | | -(**void**)ApplyWithENTUserID:(NSString \*)userID                  User\_name:(NSString \*)user\_name               legal\_person: (NSString \*)legal\_person            ent\_register\_no:(NSString \*)ent\_register\_no                   card\_num:(NSString \*)card\_num                  phone\_num:(NSString \*)phone\_num                 user\_email:(NSString \*)user\_email                    cert\_ext2:(NSString \*)cert\_ext2                  cert\_ext3:(NSString \*)cert\_ext3                  cert\_ext4:(NSString \*)cert\_ext4                   delegate:(**id**<CertDelegate>)delegate; | | |
| 企业证书申请实现方法 |  | | [[AXUserInfo sharedInstance] ApplyWithENT UserID:@“111” User\_name:@"CA" legal\_person:@"张三" ent\_register\_no:@"91320191MA1WL8B040" card\_num:@"32010319900101234" phone\_num:@"18012345678" user\_email:nil cert\_ext2:nil cert\_ext3:nil cert\_ext4:nil delegate:self];  (说明: 需使用代理方法CertDelegate  获取回调证书信息，代理方法-(void)CertDictionary:(NSDictionary \*)dictionary | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| user\_name | NSString | | Y | 企业名称 |
| card\_num | NSString | | N | 法人身份证号 |
| phone\_num | NSString | | N | 法人联系方式 |
| legal\_person | NSSting | | N | 法人姓名 |
| ent\_register\_no | NSSting | | N | 统一社会代码号 |
| user\_email | NSString | | N | 企业邮箱 |
| cert\_ext2 | NSString | | N | 扩展项2（证书的用户额外信息项，没有则不传） |
| cert\_ext3 | NSString | | N | 扩展项3（证书的用户额外信息项，没有则不传） |
| cert\_ext4 | NSString | | N | 扩展项4（证书的用户额外信息项，没有则不传） |
| Delegate | Delegate | | Y | 回调方法 |

## 虚拟申请证书接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现虚拟证书的在线申请 | | |
| 企业证书申请函数原型 |  | | -(**void**)ApplyWithVHUserID:(NSString \*)userID                 User\_name:(NSString \*)user\_name                   cert\_ou: (NSString \*)cert\_ou                    cert\_o:(NSString \*)cert\_o                    cert\_s:(NSString \*)cert\_s                    cert\_l:(NSString \*)cert\_l                    cert\_e:(NSString \*)cert\_e                 cert\_ext2:(NSString \*)cert\_ext2                 cert\_ext3:(NSString \*)cert\_ext3                 cert\_ext4:(NSString \*)cert\_ext4                  delegate:(**id**<CertDelegate>)delegate; | | |
| 企业证书申请实现方法 |  | | [[AXUserInfo sharedInstance] ApplyWithVHUserID：@“111” User\_name:@"张三" cert\_ou:nil cert\_o:@"cert\_o" cert\_s:nil cert\_l:nil cert\_e:nil cert\_ext2:nil cert\_ext3:nil cert\_ext4:nil delegate:self];  (说明1:cert\_category为企业申请证书的标识为固定参数”VH”)  (说明2: 需使用代理方法Certdelegate获取回调证书信息，代理方法-(void)certInfo:(NSDictionary \*)dic  ((说明: 需使用代理方法CertDelegate  获取回调证书信息，代理方法-(void)CertDictionary:(NSDictionary \*)dictionary | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| user\_name | NSString | | Y | 用户名 |
| cert\_ou | NSString | | N | Cert\_ou（用户信息项） |
| cert\_s | NSString | | N | Cert\_s（用户信息项） |
| cert\_l | NSSting | | N | Cert\_l（用户信息项） |
| cert\_e | NSSting | | N | Cert\_e（用户信息项） |
| cert\_o | NSString | | Y | Cert\_o（注意：此项为必传项） |
| cert\_ext2 | NSString | | N | 扩展项2（证书的用户额外信息项，没有则不传） |
| cert\_ext3 | NSString | | N | 扩展项3（证书的用户额外信息项，没有则不传） |
| cert\_ext4 | NSString | | N | 扩展项4（证书的用户额外信息项，没有则不传） |
| Delegate | Delegate | | Y | 回调方法 |

## 原文签名接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现私钥签名 | | |
| 函数原型 |  | | -(**void**)passSignWithUserID:(NSString \*)userID  data:(NSString \*)data  pn:(NSString \*)pn                  data\_type:(DataType)data\_type                    success:(CallBack)success                      error:(ErrorBack)error; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data | NSString | | Y | 签名原文 |
| pn | NSString | | N | 项目编号（区分PIN码缓存项目） |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 原文验签接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现公钥验签 | | |
| 函数原型 |  | | - -(**void**)VerifyWithUserID:(NSString \*)userID                     data:(NSString \*)data                sign\_data:(NSString \*)sign\_data                data\_type:(DataType)data\_type                  success:(CallBack)success                    error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data | NSString | | Y | 签名原文 |
| sign\_data | NSString | | Y | 签名结果（签名后返回的参数） |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 原文加密接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现公钥加密 | | |
| 函数原型 |  | | -(**void**)EncryptWithUserID:(NSString \*)userID                      data:(NSString \*)data                 data\_type:(DataType)data\_type                   success:(CallBack)success                     error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data | NSString | | Y | 加密原文 |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 原文解密接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现私钥解密 | | |
| 函数原型 |  | | -(**void**)passDecryptWithUserID:(NSString \*)userID                          data:(NSString \*)data                     data\_type:(DataType)data\_type                       success:(CallBack)success                         error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| data | NSString | | Y | 加密结果 |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 重置PIN码接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，实现重置PIN码 | | |
| 函数原型 |  | | -(**void**)PINRestWithUserID:(NSString \*)userID                 user\_name:(NSString \*)user\_name           ent\_register\_no:(NSString \*)ent\_register\_no                 card\_type:(CardType)card\_type                  card\_num:(NSString \*)card\_num                 phone\_num:(NSString \*)phone\_num; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| user\_name | NSString | | Y | 用户名/企业名称 |
| ent\_register\_no | NSString | | N | 企业税号 |
| card\_num | NSString | | Y | 证件号 |
| card\_type | CardType | | Y | /\*\* 身份证 \*/      IDCard,      /\*\* 临时身份证 \*/      InterimIDCard,       /\*\* 户口本 \*/      HouseholdRegister,       /\*\* 护照 \*/      Passport,       /\*\* 军人身份证 \*/      SoldierIDCard,       /\*\* 武警身份证 \*/      ArmedPoliceIDCard, |
| phone\_num | NSString | | Y | 电话号码 |
|  | | 说明：该接口调用会弹出重置PIN码界面 | | |

## 更新证书接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，更新证书 | | |
| 函数原型 |  | | -(**void**)CertUpdateWithUserID:(NSString \*)userID                    user\_name:(NSString \*)user\_name                    phone\_num:(NSString \*)phone\_num     user\_city:(NSString \*)user\_city                      user\_email:(NSString \*)user\_email                      success:(CallBack)success                        error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| user\_name | NSString | | Y | 用户名/企业名称(必传)/证书名称（必传） |
| phone\_num | NSString | | N | 电话号码 |
| user\_city | NSString | | N | 城市 |
| user\_email | NSString | | N | 邮箱 |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 查询证书接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，查询证书 | | |
| 函数原型 |  | | -(**void**)QueryCertInfoWithUserID:(NSString \*)userID                         success:(CallBack)success                           error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 变更证书状态接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，变更证书状态 | | |
| 函数原型 |  | | -(**void**)CertStatusUpdateWithUserID:(NSString \*)userID                        update\_type:(CertStatus)update\_type                            success:(CallBack)success                              error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| update\_type | CertStatus | | Y | Freeze：冻结；Unfreeze：解冻；Revoke：废除； |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 延期证书接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，延期证书 | | |
| 函数原型 |  | | -(**void**)certPostponeWithUserID:(NSString \*)userID                        success:(CallBack)success                          error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 重置缓存密码时间接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，重置缓存密码时间 | | |
| 函数原型 |  | | -(**void**)CachePriKeyWithUserID:(NSString \*)userID  pn:(NSString \*)pn                       success:(CallBack)success                         error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| pn | NSString | | N | 项目编号（区分PIN码缓存项目） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 证书签章接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，进行证书签章 | | |
| 函数原型 |  | | -(**void**)PassCertSealWithUserID:(NSString \*)userID                           data:(NSString \*)data                             pn:(NSString \*)pn                      data\_type:(DataType)data\_type                        success:(CallBack)success                          error:(ErrorBack)err | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data | NSString | | Y | 签章数据 |
| pn | NSString | | N | 项目编号（区分PIN码缓存项目） |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 证书登录接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，进行证书登录 | | |
| 函数原型 |  | | -(**void**)PassLoginMobileWithUserID:(NSString \*)userID  data:(NSString \*)data  pn:(NSString \*)pn                         data\_type:(DataType)data\_type                           success:(CallBack)success                             error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| data | NSString | | Y | 签名数据 |
| pn | NSString | | N | 项目编号（区分PIN码缓存项目） |
| data\_type | DataType | | Y | 待签名数据类型：Data16System表示data是原文数据转成16进制，DataByte表示data是byte类型数据（ 16进制编码），Date16CN表示data是中文转成16进制，DataInitial表示data是原文数据，DataBase64表示 data是byte类型数据（ Base64编码加密后的值） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 获取二维码解绑接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，获取二维码数据 | | |
| 函数原型 |  | | -(**void**)GetQrcodeWithUserID:(NSString \*)userID                     success:(CallBack)success                       error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 扫码解绑接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，进行扫码解绑 | | |
| 函数原型 |  | | -(**void**)UntiePhoneWithUserID:(NSString \*)userID                         QCCode:(NSString \*)QCCodeString                     delegate:(**id**<CertDelegate>)delegate; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |
| QCCode | NSString | | Y | 二维码数据 |
| delegate | delegate | | Y | 回调对象 |

## 清除本地证书

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，删除手机盾存储的证书信息 | | |
| 函数原型 |  | | -(**BOOL**)clearCertinfoWithUserID:(NSString \*)userID; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |

## 签名图片数据存储

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，获取数据 | | |
| 函数原型 |  | | -(**void**)passCertSignAndPicOrSignWithUserID:(NSString \*)userID                      data\_base64:(NSString \*)data\_base64                            success:(CallBack)success                              error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识（签名图片ID） |
|  | data\_base64 | NSString | | Y | type为2时表示签名图片base64编码字符串 |
|  | success | CallBack | | Y | 回调结果 |
|  | error | ErrorBack | | Y | 回调结果 |

## 手写面板签名图片数据存储

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，进行手写签名储存到服务器 | | |
| 函数原型 |  | | -(**void**)ReturnServerSignatureImageWithUserID:(NSString \*)userID                                      success:(CallBack)success                                        error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识（签名图片ID） |
|  | success | CallBack | | Y | 回调结果 |
|  | error | ErrorBack | | Y | 回调结果 |

## 获取签名图片数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，获取数据 | | |
| 函数原型 |  | | -(**void**)GetCertSignAndPicOrSignWithUserID:(NSString \*)userID                                success:(CallBack)success                                  error:(ErrorBack)err; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识（签名图片ID） |
| success | CallBack | | Y | 回调结果 |
| error | ErrorBack | | Y | 回调结果 |

## 手写签名面板获取图片

|  |  |  |
| --- | --- | --- |
| 功能描述 |  | 业务端通过调用该接口，手写签名获取图片 |
| 函数原型 |  | -(**void**)ReturnSignImage:(**void** (^)(UIImage \*))Image;  （该接口直接返回手写图片） |

## 判断证书是否存在

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，判断证书是否存在 | | |
| 函数原型 |  | | -(**BOOL**)isExistCertWithUserID:(NSString \*)userID; | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |

## 获取本地证书信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能描述 |  | | 业务端通过调用该接口，获取本地证书信息 | | |
| 函数原型 |  | | -(NSDictionary \*)GetCertDictionaryWithUserID:(NSString \*)userID;  （说明：如果有证书则返回证书信息，没有证书则返回空） | | |
| 参数 | 参数名 | 数据类型 | | 是否必填 | 备注 |
| userID | NSString | | Y | 用户唯一标识 |

## 获取设备标识

|  |  |  |
| --- | --- | --- |
| 功能描述 |  | 业务端通过调用该接口，返回设备标识 |
| 函数原型 |  | +(NSString \*)GetPhone\_IDFV; |

# 4.附录说明：

1.申请证书接口分为个人证书申请、企业证书申请和虚拟证书申请，添加协议CertDelegate；

**协议方法:** -(void)CertDictionary:(NSDictionary \*)dictionary；

2.在调用其他接口前可先判断证书是否存在来做其他的逻辑处理；

**证书信息**参数说明如下表：

|  |  |  |
| --- | --- | --- |
| 参数 | 参数类型 | 参数说明 |
| PINValue | NSString | 申请证书时设置的PIN码 |
| encCert | NSString | 加密证书 |
| encCertInfo | NSDictionary | 加密证书信息 |
| lraInfo | NSString | 证书安装码 |
| signCert | NSString | 签名证书 |
| signCertInfo | NSDictionary | 签名证书信息 |
| uniqueId | NSString | 证书唯一标识 |

部分接口返回码说明：

|  |  |  |
| --- | --- | --- |
| 返回码 | 说明 | 备注 |
| 1 | 成功 | - |
| 2001 | UserID不存在 |  |
| 3001 | 证书已存在 | 申请证书时如果本地证书存在则返回 |
| 3002 | 证书信息不存在 | 调用接口时如果证书失效或者本地证书被清除（无证书）则返回 |
| 3003 | 二维码信息错误 |  |
| 3004 | type值错误 |  |
| 3005 | 证件类型错误 |  |
| 3006 | data\_type类型错误 |  |
| 3007 | 证书状态类型错误 |  |
| 3008 | 网络连接或数据类型处理错误 |  |
| 3009 | 证书类型错误 |  |