

集成文档-iOS

创蓝实人认证&活体检测 SDK 集成文档 V2.0.0.X

合规性说明

升级指南

1.2.0.0.4以下版本升级到2.0.0.4+版本

一、SDK集成

1、获取SDK

2、开发环境搭建

2.1 导入FrameWork

2.2 配置Xcode

3、SDK 初始化

3.1 初始化

4、调用活体检测功能

4.1 回调数据说明

4.2调用流程

5、调用身份证 OCR 功能

5.1 api如下

6、API功能模块说明

6.1 如果只用到活体或者OCR功能

6.1 如果用到活体和OCR功能

二、历史版本

创蓝实人认证&活体检测 SDK 集成文档 V2.0.0.X

名称： 实人认证&活体检测 iOS SDK 集成文档

版本： V2.0.0.X

作者： 王亚东

更新日期： 2021-12-22

邮箱: wangyd@253.com

手机: 16621001237

合规性说明

SDK名称	场景描述	收集个人信息的类型	第三方机构名称	数据是否加密传输	第三方隐私政策链接
实名认证SDK	活体检测	实人认证&活体检测SDK提供活体检测服务过程中收集和使用的信息包括: 用户的面部识别信息、用户的设备摄像头权限: 用于活体检测服务中人脸检测、人脸检索、人脸比对;	上海创蓝云智信息科技股份有限公司	是	http://git.253.com/wanggang/sdk-release-package

升级指南

1.2.0.0.4以下版本升级到2.0.0.4+版本

2.0.0.4版本将SDK拆分为两个SDK, 多出OCRSDK, 需要单独设置token, 如下:

Objective-C | 复制代码

```
1 [CL0CRManager setToken:@" token "];
```

一、SDK集成

说明

- SDK使用Xcode 12.5.1打包, 支持iOS9.0及以上系统版本。
- 2.0.0.2版本开始使用Xcode13打包

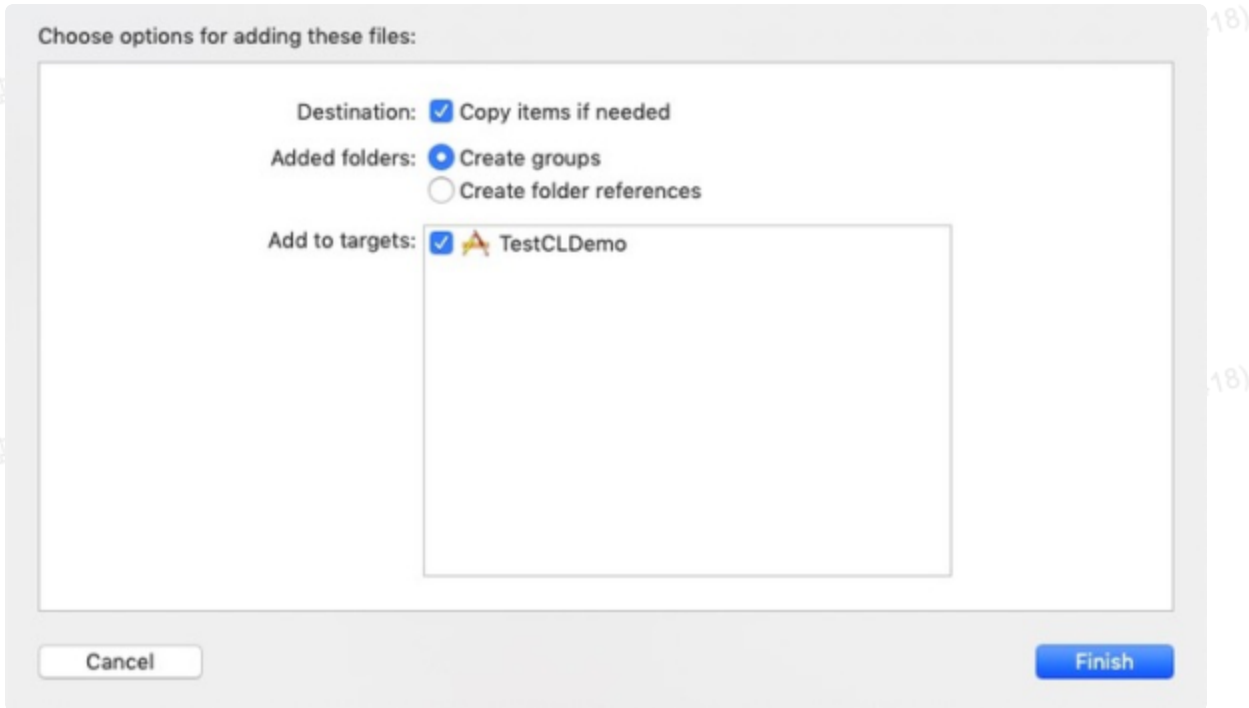
1、获取SDK

从官网[下载](#)活体检测 sdk 包。

2、开发环境搭建

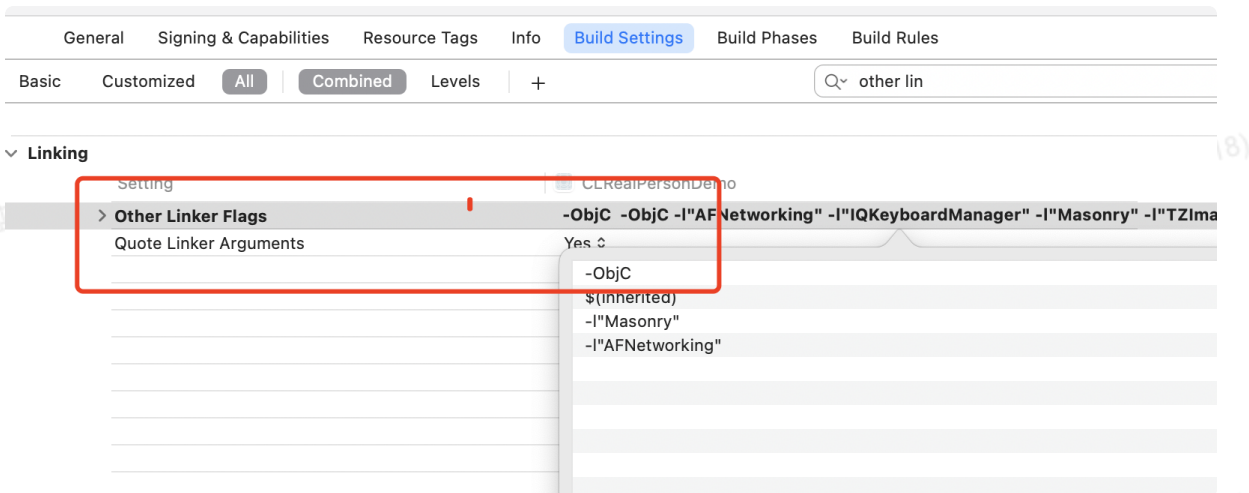
2.1 导入FrameWork

将实人认证SDK压缩包中framework中的所有资源添加到工程中，并选择 Copy Items if need 选项

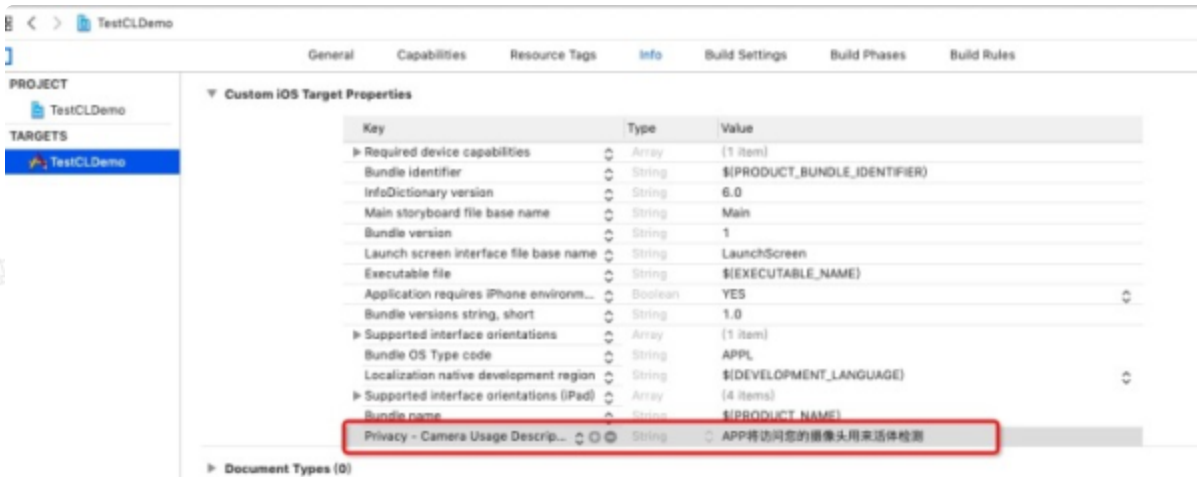


2.2 配置Xcode

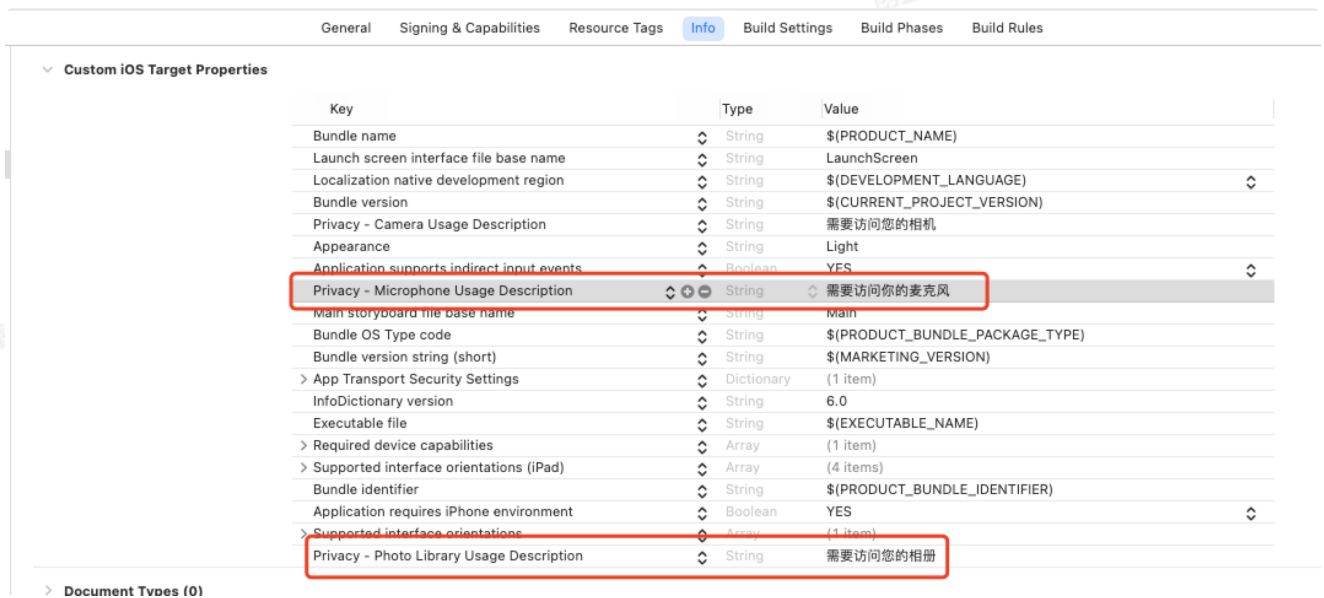
- Xcode→TARGETS→Build Settings→Linking→Other Linker Flags 添加-ObjC



- Xcode→TARGETS→Info添加相机权限访问描述



- 如果OCR需要用相册选择身份证或者在活体中需要麦克风播放提示音则需在Xcode->TARGETS->Info添加相册或者麦克风权限访问描述；（该权限为demo所用权限非SDK必须权限，仅供参考）



3、SDK 初始化

3.1 初始化

建议在 **Application** 的 `didFinishLaunchingWithOptions` 方法中进行初始化：

```
1 ▾ - (BOOL)application:(UIApplication *)application
    didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
2     ...
3
4     // 控制台log开关
5     [CLCoreManager printConsoleEnable:YES];
6     // 设置token, 建议在需要使用文档中4, 5功能时设置, 有效期30分钟
7     [CLRealPersonManager setToken:@" token "];
8     // 2.0.0.4版本拆分ocr, 需要单独设置token
9     [CLOCRManager setToken:@" token "];
10
11
12     // 2.0.0.2之前版本使用:
13     [CLCoreManager initWithAppid:@" appid "];
14     // 2.0.0.2及之后版本使用:
15     [CLCoreManager initWithAppkey:@" appkey "];
16
17     ...
18     return YES;
19 }
```

注意：在 initWithAppid: 中设置 **appid**：和在 setToken: 中设置 token。

温馨提示：

appid/appkey 查看路径，登录开发者平台，右上角点击「应用管理」找到自己的应用，点击后面的「配置」按钮，然后可以看到 **appkey**（**对应SDK中appid、appkey**）了，点击复制按钮即可。token 从服务端api获取

4、调用活体检测功能

4.1 回调数据说明

① 会在 completionHandler中回调 CLStatus 信息，具体的信息如下表所示：

CLStatus	错误信息	备注说明
CLCheckPass	活体检测通过	/
CLCheckNotPass	活体检测不通过	/
CLOperationTimeout	操作超时，用户未在规定时间内完成动作	/
CLGetConfFaild	获取配置信息失败	/
CLRecordFaild	视频录制失败	/
CLOnlineCheckFaild	请求检测结果失败	/
CLOnlineUploadFaild	上传图片失败	/
CLSDKError	SDK内部发生错误	/
CLRPVedioNoMate	人脸动作与提示动作不吻合，请重试	/
CLRPVedioNoFace	没有检测到人脸，请重试	/
CLRPVedioMoreFace	检测到有多张人脸，请重试	/
CLRPVedioIllumination	检测到光线不足，请重试	/
CLRPVedioAngle	检测到人脸角度有偏差，请重试	/
CLRPVedioBlue	检测到人脸模糊，请重试	/
CLRPVedioMask	检测到人脸有遮挡，请重试	/
CLRPVedioSmall	检测到人脸过于靠近屏幕，请重试	/
CLNetError	无网络链接	/
CLRPVedioError	其他错误	/
CLRPNoMoney	账号余额不足	V2.0.0.3新增

温馨提示：

① 调用活体检测方法之前，请确保已经获取到相机、！！！！

② 我们提供了默认的声音文件和 GIF 图片文件，如不满意可以替换，具体逻辑在demo中实现，可任意更改；

4.2调用流程

如下从4.2.1到4.2.4时一个完整的检测流程

4.2.1 先初始化活体检测对象

```
1  /// 初始化活体检测对象
2  /// @param recordView          传入放置检测活体的recordView对象
3  - (instancetype)initWithRecordView:(UIView *)recordView;
```

4.2.2 在开始活体检测前进行相关参数设置

```
1  /// 设置活体检测的超时时间（在开始活体检测之前设置）
2  /// @param timeout              请传入10-120范围内的时间值，单位s
3  - (void)setTimeoutInterval:(NSTimeInterval)timeout;
4
5  /// 设置活体检测每个动作的时长（在开始活体检测之前设置）
6  /// @param duration             每个动作的时长（1-5s）
7  - (void)setActionDuration:(CGFloat)duration;
8
9  /// 设置活体检测动作的个数（在开始活体检测之前设置，默认为1-3的随机数 V2.0.0.3+版本
   默认为1个）
10 /// @param number               动作的个数（1-3）
11 /// @param securityLevel        安全级别：默认为低（随机动作数），高至少为两
   个动作（会修正number设置数量）
12 - (void)setActionNumber:(NSInteger)number securityLevel:
   (CLSecurityLevel)securityLevel;
13
14 /// 设置检测时脸离屏幕距离（在开始活体检测之前设置）V2.0.0.3版本增加
15 /// @param distance             默认0（-1：稍近，0：正常，1：稍远）
16 - (void)setDistanceNumber:(NSInteger)distance;
```

4.2.3 开始活体检测

```

1  /// 开始活体检测
2  /// @param actionsHandler      活体检测动作序列号的回调（非主线程）
3  /// @param completionHandler  活体检测结果的回调（非主线程），结果状态见
    CLStatus枚举类型
4  - (void)startLiveDetectWithActionsHandler:
    (CLAcitionsHandler)actionsHandler
5      completionHandler:
    (CLCompletionHandler)completionHandler;
6
7  /// 活体检测，视频转发
8  /// @param urlString           视频转发之后URL或者
    startLiveDetectWithActionsHandler: 返回的url
9  /// @param isReTran           表示是否转发，YES: 转发urlString参数值为转
    发后的url NO: 不转发urlString参数值为startLiveDetectWithActionsHandler: 中返回
    的url
10 - (void)resultWithDetectUrl:(NSString *)urlString
11      isReTran:(BOOL)isReTran;

```

4.2.4 停止活体检测

```

1  /// 停止活体检测
2  /// 调用时机:
3  ///      1、在活体检测结果的回调里调用
4  ///      2、未完成活体检测，需要中止时调用（超时不要调用）
5  - (void)stopLiveDetect;

```

5、调用身份证 OCR 功能

身份证 OCR 接口封装了常用 OCR 的使用场景，例如，身份证单面 OCR，身份证双面 OCR，身份证二要素认证等类型，可根据自己的实际需求，选择合适的调用类型。

5.1 api如下


```

1  /// 身份证OCR接口      △ 先调用人像面（图片大小建议压缩为1M以内，超过会被SDK压缩）
2  /// @param image          身份证图片
3  /// @param isFront        YES:为人像面 NO:为国徽面
4  /// @param complecte      结果回调（非主线程） data数据为字典，格式如下：
5
6                          /*
7                          国徽面数据： {
8                              "expire_date"      = "失效日期";
9                              "issuing_authority" = "签发机关";
10                             "issuing_date"      = "签发日期";
11                             };
12                             人像面数据： {
13                                 "address"       = "地址";
14                                 "brith_day"     = "出生日期";
15                                 "id_card_no"    = "身份证号";
16                                 "name"          = "姓名";
17                                 "nation"        = "民族";
18                                 "sex"           = "性别";
19                             };
20                             */
21 + (void)realPersonOCR:(UIImage *)image isFront:(BOOL)isFront complect:
22 (CLCoreNetComplete)complecte;
23
24 /// 身份证OCR接口      识别人像面和国徽面（单图片大小建议压缩为1M以内，超过会被SDK压缩）
25 /// @param frontImage     人像面照片
26 /// @param backImage      国徽面照片
27 /// @param complecte      结果回调（非主线程） data数据为字典，格式如下：
28
29                          /* {
30                              back = {
31                                  "expire_date"      = "失效日期";
32                                  "issuing_authority" = "签发机关";
33                                  "issuing_date"      = "签发日期";
34                              };
35                              front = {
36                                  "address"       = "地址";
37                                  "brith_day"     = "出生日期";

```

```

36                                     "id_card_no" = "身份证号";
37                                     "name"      = "姓名";
38                                     "nation"    = "民族";
39                                     "sex"       = "性别";
40                                     };
41                                 }
42                             */
43 + (void)realPersonOCR:(UIImage *)frontImage backImage:(UIImage
*)backImage complect:(CLCoreNetComplete)complecte;
44
45
46 /// 身份证认证
47 /// @param name          姓名
48 /// @param idNumber      身份证号码
49 /// @param complecte     结果回调（非主线程） data数据为字典，格式如
下：
50                             /*{
51                                 "order_no"   : "业务唯一流水号",
52                                 "city"       : "城市",
53                                 "country"    : "县区",
54                                 "gender"     : "性别：1：男、2：女",
55                                 "age"        : "年龄",
56                                 "remark"     : "备注，例：一致",
57                                 "birthday"   : "生日，格式是
yyyyMMdd",
58                                 "result"     : "返回结果： 01-认证一
致(收费) 02-认证不一致(收费) 03-认证不确定(收费) 04-认证失败(不收费)",
59                                 "handle_time" : "查询时间 例：2018-
04-09 15:05:01",
60                                 "province"  : "省份"
61                             }*/
62 + (void)realPersonDectect:(NSString *)name idNumber:(NSString *)idNumber
complect:(CLCoreNetComplete)complecte;
63

```

6、API功能模块说明

6.1 如果只用到活体或者OCR功能

先调用3中SDK初始化，然后调用4或者5中对应的api功能

6.1 如果用到活体和OCR功能

先调用3中SDK初始化，然后调用4和5中对应的api功能

具体逻辑参考demo，demo提供了具体的UI和逻辑实现。

二、历史版本

发布日期	发布版本	更新说明
2021-07-05	V1.0.0.0	初版发布
2021-07-12	V1.0.0.1	优化SDK包体积
2021-07-29	V1.0.0.2	新增身份证双面OCR接口，优化OCR接口安全性
2021-10-11	V2.0.0.0	优化SDK包体积，更换为API形式验证
2021-12-10	V2.0.0.1	优化文档和注释，demo中转发逻辑调整
2021-12-22	V2.0.0.2	更改appid为appkey，demo适配iOS15
2022-03-18	V2.0.0.3	优化SDK功能逻辑
2022-04-11	V2.0.0.4	拆分SDK