

【版权声明】

【商标声明】

【免责声明】

【修订记录】

1. 引言

1.1 编写目的

1.2 文档使用说明

2. 概述

3. 接入流程

3.1 依赖环境

3.2 集成步骤

3.2.1 手动接入方式

3.2.2 使用Pod方式接入

3.2.3 权限设置

4. API详细概述

4.1 HuiYanPrivateApi

4.1.1 release()

4.1.2 sdkVersion()

4.1.3 startGetAuthConfigData()

4.1.4 startAuthByLiveData()

4.1.5 StopAuthActionErr:

4.2 HuiYanPrivateConfig

4.3 PrivateLiveDataEntity

4.4 PrivateCompareResult

4.4.1 加密字段取值说明

4.5 PrivateGetConfigResult

4.6 HuiYanConfigSuccCallback

4.7 HuiYanConfigFailCallback

4.8 HuiYanResultSuccCallback

4.9 HuiYanResultFailCallback

4.10 HY_CameraPreset

4.11 HuiYanPrivateDelegate

4.12 HYEncryptModel

4.13 HYAuthTipsEvent

4.14 错误码

4.15 CameraZoom

4.16 HuiYanScreenBrightnessApi

4.17 HY_LanguageType

4.18 HYShowTimeOutMode

5. SDK接口使用说明

5.1 初始化配置，并拉取配置参数

5.2 完成剩余步骤活体核身

5.3 SDK资源释放

5.4 常见问题与解决方法

【版权声明】

本文档著作权归腾讯云计算（北京）有限责任公司（以下简称“腾讯云”）单独所有，未经腾讯云事先书面许可，任何主体不得以任何方式或理由使用本文档，包括但不限于复制、修改、传播、公开、剽窃全部或部分本文档内容。

本文档及其所含内容均属腾讯云内部资料，并且仅供腾讯云指定的主体查看。如果您非经腾讯云授权而获得本文档的全部或部分内容，敬请予以删除，切勿以复制、披露、传播等任何方式使用本文档或其任何内容，亦请切勿依本文档或其任何内容而采取任何行动。

【商标声明】



“腾讯”、“腾讯云”及其它腾讯云服务相关的商标、标识等均为腾讯云及其关联公司各自所有。若本文档涉及第三方主体的商标，则应依法由其权利人所有。

【免责声明】

本文档旨在向客户介绍本文档撰写时，腾讯云相关产品、服务的当时的整体概况，部分产品或服务在后续可能因技术调整或项目设计等任何原因，导致其服务内容、标准等有所调整。因此，本文档仅供参考，腾讯云不对其准确性、适用性或完整性等做任何保证。您所购买、使用的腾讯云产品、服务的种类、内容、服务标准等，应以您和腾讯云之间签署的合同约定为准，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【修订记录】

编号	修改日期	修改人	更新内容
01	2021-6-23	jerrydong	V1.0.6发布
02	2021-7-14	jerrydong	V1.0.7发布
03	2021-12-02	jerrydong	V1.0.8发布
04	2022-01-10	jerrydong	V1.0.8.2发布
05	2022-01-20	jerrydong	v1.0.9发布
06	2022-02-09	jerrydong	v1.0.9.2发布
07	2022-06-20	Jerrydong	v1.0.9.10发布
08	2022-07-20	jerrydong	v1.0.9.12发布
09	2022-09-26	jerrydong	v1.0.9.14发布
10	2022-12-30	webertzhang	v1.1.1.1发布
11	2023-1-2	webertzhang	v1.1.1.2发布
12	2023-1-13	webertzhang	v1.1.1.3b发布
13	2023-1-29	webertzhang	v1.1.1.4发布
14	2023-3-23	webertzhang	v1.1.1.9发布
15	2023-4-20	webertzhang	v1.1.1.10发布
16	2023-4-27	webertzhang	v1.1.1.12发布
17	2023-5-05	webertzhang	v1.1.1.13发布

1. 引言

1.1 编写目的

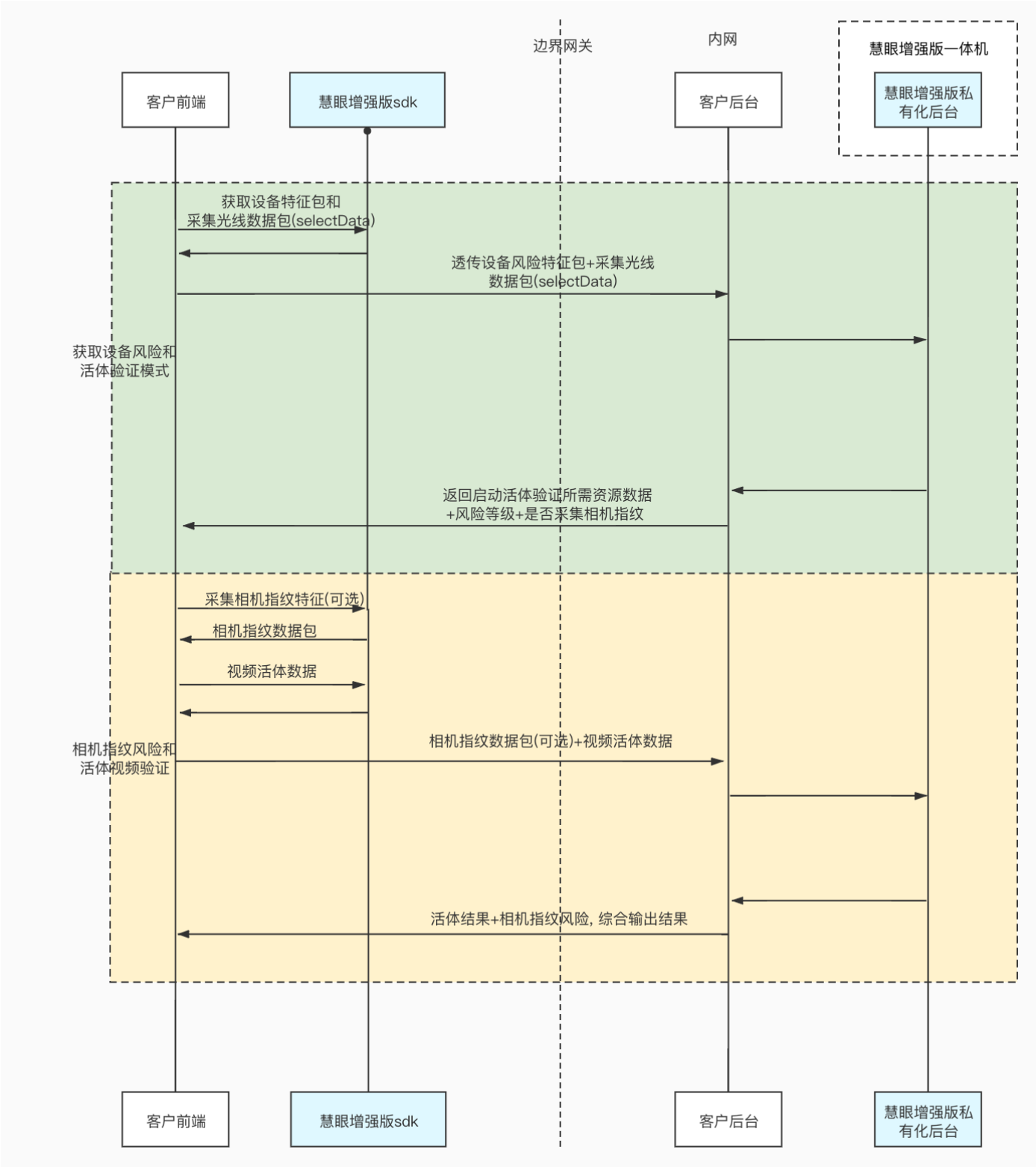
本文档对慧眼私有化产品iOS版本提供了对接指导，列明了产品主要功能的API方法以及调用方式说明。

1.2 文档使用说明

本文档适用于所有使用慧眼私有化iOS版的用户。

2. 概述

慧眼私有化部署的总体流程如下图所示，主要展示了核身SDK、客户APP以及客户后台与慧眼私有化一体机的整体交互流程：



3. 接入流程

本节主要介绍如何将慧眼私有化iOS端核身SDK集成到您的app当中，iOS端SDK主要封装了人脸的活体检测功能。

开发准备： 提供bundle Id并与销售人员申请获取License文件。

3.1 依赖环境

1. 开发环境 Xcode 11.0 及以上版本
2. 慧眼iOS SDK 适用于iOS9.0及以上版本

3.2 集成步骤

3.2.1 手动接入方式

1. 导入相关库及文件

Link Binary With Libraries导入相关Framework SDK依赖的库如下：

```
├—HuiYanSDK.framework
├—YTCv.framework
├—YTSm.framework
├—libRiskFeatures.a
├—tnnliveness.framework
├—YTCommonLiveness.framework
├—YTFaceAlignmentTinyLiveness.framework
├—YTFaceDetectorLiveness.framework
├—YTFaceLiveReflect.framework
├—YTFaceTrackerLiveness.framework
├—YTPoseDetector.framework
├—YtSDKKitActionLiveness.framework
├—YtSDKKitFramework.framework
├—YtSDKKitFrameworkTool.framework
├—YtSDKKitReflectLiveness.framework
├—YtSDKKitSilentLiveness.framework
//若未使用国密加密功能，以下库可不必添加
├—YtTencentSM
├—libgmp.a
├—libtasn1.a
├—libTencentSM.a
```

2. Link Binary With Libraries导入系统Framework

```
├— Accelerate.framework
```

3. Copy Bundle Resources中导入bundle文件

```
└─ face-tracker-v003.bundle
└─ HuiYanSDKUI.bundle
```

3.2.2 使用Pod方式接入

1. 将CloudHuiYanSDK_FW文件夹复制到集成项目Podfile同级目录下
2. 在Podfile设置

```
target 'HuiYanAuthDemo' do
  use_frameworks!
  pod 'CloudHuiYanSDK_FW', :path => './CloudHuiYanSDK_FW'
end
```

3. pod install 更新

文件层级和具体的设置可以参考Demo

4. Build Phases设置

1. Other Linker Flags 新增 **-ObjC**
2. 接入.m文件 设置后缀为.mm

3.2.3 权限设置

SDK需要摄像头使用权限，请添加对应的权限声明。在主项目info.plist 配置中添加下面key-value值

```
<key>Privacy - Camera Usage Description</key>
<string>人脸核身需要开启您的摄像头权限，用于人脸识别</string>
```

4. API详细概述

慧眼SDK主要涉及如下几个类，它们分别是API的接口类HuiYanPrivateApi， 参数配置类HuiYanPrivateConfig， 结果回调类HuiYanConfigCallback以及HuiYanResultCallBack。

4.1 HuiYanPrivateApi

API	功能描述
release()	资源释放接口
sdkVersion()	慧眼SDK版本号
startGetAuthConfigData	获取慧眼SDK本地配置信息的接口
startAuthByLightData()	传入服务器获取的光线序列， 继续完成活体核身检测的接口
stopAuthActionErr	当获取后端获取光线数据错误时， 客户需要主动停止时调用

4.1.1 release()

```
+ (void)release;
```

功能介绍：

慧眼SDK资源释放的接口。

4.1.2 sdkVersion()

```
/// 获取SDK版本号  
+ (NSString *_Nonnull)sdkVersion;
```

功能介绍：

慧眼SDK版本号。

4.1.3 startGetAuthConfigData()

```
/**  
 * 初始化获取设备信息  
 *  
 * @param privateConfig sdk配置类  
 * @param huiYanConfigSuccCallback 配置成功，数据回调  
 * @param huiYanConfigFailCallback 配置失败回调  
 */  
+ (void)startGetAuthConfigData:(HuiYanPrivateConfig *)privateConfig  
    withSuccCallback:(HuiYanConfigSuccCallback)huiYanConfigSuccCallback  
    withFialCallback:(HuiYanConfigFailCallback)huiYanConfigFailCallback;
```

功能介绍：

本地检测慧眼SDK同时拉取配置参数，用来作为后续兑换光线序列的参数。

传入参数：

参数类型	参数名称	参数含义
HuiYanPrivateConfig	huiYanPrivateConfig	配置的参数
HuiYanConfigSuccCallback	huiYanConfigSuccCallback	拉取配置结果成功的回调
HuiYanConfigFailCallback	huiYanConfigFailCallback	拉取配置结果失败的回调

4.1.4 startAuthByLiveData()

```
/**
 * 开始活体验证
 *
 * @param liveDataEntity 光线数据类
 * @param huiYanResultSuccCallback 验证成功，数据回调
 * @param huiYanResultFailCallback 验证失败回调
 */
+ (void)startAuthByLiveData:(PrivateLiveDataEntity *)liveDataEntity
    withSuccCallback:(HuiYanResultSuccCallback)huiYanResultSuccCallback
    withFialCallback:(HuiYanResultFailCallback)huiYanResultFailCallback;
```

功能介绍：

将从服务器拉取到的光线序列数据，传入慧眼SDK同时继续核身流程，并获取本地检测的结果。

传入参数：

参数类型	参数名称	参数含义
PrivateLiveDataEntity	liveDataEntity	活体数据类
HuiYanResultSuccCallback	huiYanResultSuccCallback	本地核身的结果成功回调
HuiYanResultFailCallback	huiYanResultFailCallback	本地核身的结果失败回调

4.1.5 StopAuthActionErr:

当获取后端光线序列数据发生错误时(网络或接口错误时)，调用此方法可主动关闭SDK页面，注意：**startGetAuthConfigData startAuthByLiveData** 这两个方法调用回调失败不必调用此方法，SDK会主动结束关闭页面

```
/**
 * 当获取后端光线序列数据(getLiveType接口)发生错误时(网络或接口错误)
 *
 * @param errCode 传入错误码
 * @param errMsg 传入失败原因
 */
+ (void)stopAuthActionErr:(int)errCode errMsg:(NSString *_Nonnull) errMsg;
```

功能介绍：

当获取后端光线序列数据(getLiveType接口)发生错误时(网络或接口错误时)，客户可以调用此方法可主动关闭SDK页面，在调用此方法后，会在**startGetAuthConfigData()**的**FialCallback**回调，输出 errCode errMsg，注意：**startGetAuthConfigData() startAuthByLiveData()** 这两个方法调用回调失败不必调用此方法，可以忽略此方法，SDK会主动结束关闭页面。

传入参数：

参数类型	参数名称	参数含义
int	errCode	主动传入失败错误码
NSString	errMsg	主动传入失败原因

4.2 HuiYanPrivateConfig

HuiYanPrivateConfig是在启动慧眼SDK时的配置实体类，主要包含了以下属性。

类型	名称	含义	默认值
NSString	authLicense	客户申请的用户核审授权的Licens文件路径	空
long	authTimeOutMs	设置活体检测的超时时间(v1.0.9 开始这个设置无效)	10000毫秒（10秒）
BOOL	isDeleteVideoCache	是否删除核身视频的本地缓存	YES
BOOL	isRecordVideo	是否录制验证视频	YES
HY_LanguageType	languageType	SDK提示语言	HY_DEFAULT
BOOL	isEncrypt	对活体数据是否加密	NO
HY_CameraPreset	cameraPreset	相机预览尺寸类型	PRESET_640x480
id< HuiYanPrivateDelegate >	delegate	动作回调代理(设置需要在调用侧ViewController加HuiYanPrivateDelegate代理)	nil
long	prepareTimeoutMs	准备阶段倒计时 >=3000	15000
long	actionTimeoutMs	单动作倒计时>=2000	15000
NSString	userUIBundleName	设置使用自定义打包的UI bundle 布局文件	HuiYanSDKUI
HY_LanguageType	languageType	SDK提示语言	HY_DEFAULT
BOOL	isEncrypt	对活体数据是否加密	NO
HY_CameraPreset	cameraPreset	相机预览尺寸类型	PRESET_640x480
id< HuiYanPrivateDelegate >	delegate	动作回调代理(设置需要在调用侧ViewController加HuiYanPrivateDelegate代理)	nil
long	prepareTimeoutMs	准备阶段倒计时 需>=3000	15000
long	actionTimeoutMs	单动作倒计时 需 >=2000	15000
NSString	userUIBundleName	设置使用自定义打包的UI bundle 布局文件	HuiYanSDKUI
NSString	userLanguageBundleName	设置自定义扩展语言包bundle 名称(不包含后缀)	nil
NSString	userLanguageFileName	自定义语言包 bundle里面 语言文件夹名称 不包含后缀名 .lproj 例子：en	nil
NSUInteger	feedBackErrorColor	反馈异常Tips的颜色	0xFF584C
NSUInteger	feedBackTxtColor	反馈正常Tips的颜色	0x29CC85
NSUInteger	authCircleErrorColor	动作错背景圆框的颜色	0xFF584C
NSUInteger	authCircleCorrectColor	动作正确时背景圆框的颜色	0x29CC85
NSUInteger	authLayoutBgColor	核身界面背景颜色（不设置则跟随系统，浅色模式为白色背景，深色模式为黑色背景）	-100
CGFloat	feedBackTxtSize	提示文字大小（已弃用）	17.0
UIFont	feedBackTxtFont	核身提示文字字体及大小	18
UIFont	feedbackExtraTxtFont	额外提示文字大小	14
BOOL	isShowDialog	内部Dialog是否显示	YES

HYEncryptModel	encryptModel	活体数据加密算法	AES
BOOL	isCutVideo	是否录裁剪视频	YES
BOOL	isUseBackCamera	是否使用后置摄像头（如果使用后置摄像头， Reflect Mode 或者 Action + Reflect Mode 会自动报错并终止SDK，即后置摄像头不能启用反光模式）	NO
int	bestImgPitch	周转角度	30
int	bestImgRoll	滚动角度	30
int	bestImgYaw	偏转角度	30
int	continuousNum	连续帧数控制（1~30）	1
CameraZoom	zoomType	后置摄像头缩放比例	ZOOM_1X
BOOL	isNeedAngleDetect	是否开启角度检测(人脸角度偏转将报错)	NO
BOOL	isAutoScreenBrightness	是否允许SDK自动调节屏幕亮度（关闭后才可以调用 HuiYanScreenBrightnessApi 相关控制屏幕亮度的方法）	YES
NSString	riskLicense	客户申请的图灵顿授权文件路径	nil
HYShowTimeOutMode	showTimeOutMode	设置倒计时显示模式，控制各个阶段是否显示倒计时	默认显示所有
BOOL	disableDarkMode	设置是否禁用黑夜模式	NO

TIPS：设置userUIBundleName 会加载TXYOAuthingViewController名称(名称固定为这个)的布局文件，参考demo中TXYOAuthingViewController.xib，控件绑定的事件或属性**不可删除或更改**，控件已与源码绑定，可调整控件大小位置，将打包输出的userUIBundleName.bundle 添加 **Copy Bundle Resources** 位置。

4.3 PrivateLiveDataEntity

PrivateLiveDataEntity是传入慧眼SDK活体数据实体类，主要包含了以下属性。

类型	名称	含义	默认值
NSString	liveResult	光线序列的json字符串	空
NSString	riskResult	风险结果的json字符串	空
NSString	extraInfo	客户透传数据	空

4.4 PrivateCompareResult

PrivateCompareResult是慧眼SDK结果返回实体类，主要包含了以下属性。

类型	名称	含义	默认值
int	platform	客户的版本1. iOS 2. Android	1
NSString	extraInfo	客户透传数据	空
NSString	sign	加密md5值(isEncrypt=YES有效)	空
NSString	liveData	base64采集数据(isEncrypt=YES有效)	空
NSString	unEncryptLiveData	base64采集数据(isEncrypt=NO有效)	空

4.4.1加密字段取值说明

单一模式下liveData与unEncryptLiveData字段互斥

加密模式：需调用liveCompare接口，取值字段为sign和liveData。此时SDK返回liveData的是加密数据，参数透传即可；

非加密模式：需调用 liveComparePlain 接口，取值字段为unEncryptLiveData，此时SDK返回unEncryptLiveData的是非加密数据，unEncryptLiveData对应API参数liveDataPlain，其余参数对应API接口透传即可；

4.5 PrivateGetConfigResult

PrivateGetConfigResult为慧眼私有化SDK获取到的本地数据需要发送到慧眼后台，用来拉去光线序列和兑换设备风险等信息

类型	名称	含义	默认值
NSString	selectData	当前设备需要进行活体的信息(直接透传给慧眼服务器即可)	空
NSString	envRiskData	当前设备的环境风险信息(直接透传给慧眼服务器即可)	空

此实体类对应后台API接口 getLiveType，只需要将selectData、envRiskData透传给慧眼引擎即可。

4.6 HuiYanConfigSuccCallback

初始化并且获取本地配置成功的回调

```
/**
 * 初始化成功回调
 *
 * @param getConfigResult 初始化回调数据
 */
typedef void (^HuiYanConfigSuccCallback)(PrivateGetConfigResult * _Nonnull getConfigResult);
```

4.7 HuiYanConfigFailCallback

初始化并且获取本地配置失败的回调

```
/**
 * 初始化失败回调
 *
 * @param errCode 错误码
 * @param errMsg 错误信息
 */
typedef void (^HuiYanConfigFailCallback)(int errCode, NSString * _Nonnull errMsg);
```

4.8 HuiYanResultSuccCallback

本地核身流程与本地结果获取成功的回调

```
/**
 * 活体核身成功回调
 *
 * @param compareResult 活体比对数据类
 * @param videoPath 活体视频路径
 */
typedef void (^HuiYanResultSuccCallback)(PrivateCompareResult * _Nonnull compareResult,
NSString * _Nonnull videoPath);
```

4.9 HuiYanResultFailCallback

本地核身流程与本地结果获取失败的回调

```
/**
 * 活体核身失败回调
 *
 * @param errCode 错误码
 * @param errMsg 错误信息
 */
typedef void (^HuiYanResultFailCallback)(int errCode, NSString * _Nonnull errMsg);
```

4.10 HY_CameraPreset

SDK 相机枚举

```
typedef enum : NSUInteger {
    PRESET_640x480 = 0,
    PRESET_1280x720,
} HY_CameraPreset;
```

4.11 HuiYanPrivateDelegate

```
@protocol HuiYanPrivateDelegate <NSObject>
//核身Event事件
- (void)onAuthEvent:(HYAuthEvent)actionEvent;
//核身时tips发生改变的事件通知回调
- (void)actionCallbackType:(HYAuthTipsEvent)actionType;
//当认证的主view被创建的回调
- (void)onMainViewCreate:(UIView *)authView;
//界面被回收的回调
- (void)onMainViewDestroy;
@end
```

4.12 HYEncryptModel

慧眼核身活体数据支持的加密模式

HYEncryptModel 类型	含义
HY_AES	AES 加密算法
HY_SM4	SM4国密加密

```
typedef enum : NSUInteger {
    HY_AES = 0,
    HY_SM4,
} HYEncryptModel;
```

4.13 HYAuthTipsEvent

-(void)actionCallbackType:(HYAuthTipsEvent)actionType;活体检测过程中出现的Tips提示

```
typedef enum : int {
    // 不需要回调
    NONE = 0,
    // act_open_mouth 请张张嘴
    ACT_OPEN_MOUTH,
    // act_blink 请眨眨眼
    ACT_BLINK,
    // act_nod_head 请点头
    ACT_NOD_HEAD,
    // act_shake_head 请摇头
    ACT_SHAKE_HEAD,
    // act_screen_shaking 请不要晃动
    ACT_SCREEN_SHAKING,
    // act_light_not_right 光线不合适
    ACT_LIGHT_NOT_RIGHT,
    // no_face 没有检测到人脸
    NO_FACE,
    // no_left_face 请勿遮挡左脸
    NO_LEFT_FACE,
    // no_right_face 请勿遮挡右脸
    NO_RIGHT_FACE,
    // no_chin 请勿遮挡下巴
    NO_CHINE,
    // no_mouth 请勿遮挡嘴巴
    NO_MOUTH,
    // no_nose 请勿遮挡鼻子
    NO_NOSE,
    // no_left_eye, 请勿遮挡左眼
    NO_LEFT_EYE,
```

```

    // no_right_eye 请勿遮挡右眼
    NO_RIGHT_EYE,
    // pose_keep 验证中, 请保持姿势不变
    POSE_KEEP,
    // pose_closer 请靠近一点
    POSE_CLOSER,
    // pose_farther 请离远一点
    POSE_FARTHER,
    // pose_open_eye 请睁眼
    POSE_OPEN_EYE,
    // pose_incorrect 请摆正头部姿势
    POSE_INCORRECT,
    // fl_too_many_faces 请确保框内只有一张人脸
    TOO_MANY_FACE,
    // fl_incomplete_face 请将脸移动到框内
    INCOMPLETE_FACE,
    //txy_light_low 光线过低
    LIGHT_TOO_LOW,
    //txy_light_strong 光线过强
    LIGHT_TOO_STRONG,
    //yt_face_ref_angle_detect_error 变光环节人脸角度过大
    ANGLE_DETECT_LARGE
} HYAuthTipsEvent;

```

```

//默认事件不回调
HY_NONE = 0,
//启动认证
HY_START_AUTH,
//张嘴检测
HY_OPEN_MOUTH_CHECK,
//静默检测
HY_SILENCEN_CHECK,
//眨眼检测
HY_BLINK_CHECK,
//点头检测
HY_NOD_HEAD_CHECK,
//摇头检测
HY_SHAKE_HEAD_CHECK,
//reflect-反光检测
HY_REFLECT_CHECK,

```

4.14 错误码

错误码	错误码值	错误码含义
HY_SUCCESS	0	成功
HY_INITIALIZATION_PARAMETER_EXCEPTION	210	始化参数异常
HY_BUNDLE_CONFIGURATION_EXCEPTION	211	bundle配置异常
HY_YTSDK_CONFIGURATION_EXCEPTION	212	优图配置异常
HY_PLEASE_CALL_FIRST_INIT_API	213	先调用初始化接口
HY_SDK_AUTH_FAILED	214	SDK 授权失败
HY_USER_VOLUNTARILY_CANCELED	215	用户手动取消
HY_YTSDK_LOCAL_AUTH_FAILED	216	SDK 人脸本地检测失败
HY_CAMERA_OPEN_FAIL	217	相机开启失败
HY_DONOT_SWITCH_APPS	218	请勿在核身过程中切换应用
HY_CAMEREA_PERMISSION_EXCEPTION	219	摄像头权限异常
HY_SDK_VEDIO_CUT_EXCEPTION	220	视频裁剪失败
HY_LIGHT_DATA_FORMAT_EXCEPTION	221	光线数据格式错误
HY_DETECT_TIMEOUT	222	动作检测超时
HY_LIMIT_SET_PKG_SIZE	223	超过包体设置大小限制
HY_USE_BACK_CAMERA_WITH_REFLECTIVE_ERROR	227	传入数据包含反光内容请使用前置摄像头
HY_PREPARE_TIMEOUT	300	准备阶段超时
HY_LONGCHECK_TIMEOUT	301	耗时检测超时

4.15 CameraZoom

后置摄像头缩放比例枚举

CameraZoom类型	含义
ZOOM_1X	默认正常模式
ZOOM_1_5X	1.5倍
ZOOM_2X	2.0倍

```
typedef enum : NSUInteger {
    ZOOM_1X = 0,
    ZOOM_1_5X,
    ZOOM_2X,
} CameraZoom;
```

4.16 HuiYanScreenBrightnessApi

新增外部可以控制屏幕亮度操作的API方法，只有在isAutoScreenBrightness为NO时才起效

API	功能描述
beginAdjustScreenBrightness	启动调整屏幕亮度接口，可以把屏幕调整到最亮
endAdjustScreenBrightness	停止调整屏幕亮度接口，可以把屏幕恢复到beginAdjustScreenBrightness之前的亮度
resumeAdjustScreenBrightness	恢复调整屏幕亮度接口，可以把屏幕调整到最亮
pauseAdjustScreenBrightness	暂停调整屏幕亮度接口，可以把屏幕恢复到resumeAdjustScreenBrightness之前的亮度

4.17 HY_LanguageType

SDK 语言枚举值

```
typedef enum : NSUInteger {
    HY_DEFAULT = 0, //跟随系统设置
    HY_ZH_HANS, //中文简体
    HY_EN, //英文
} HY_LanguageType;
```

4.18 HYShowTimeOutMode

SDK各阶段倒计时枚举，默认三个阶段均展示倒计时标签，需要隐藏则使用HY_TIMEOUT_HIDDEN_MODE，若需要组合阶段则showTimeOutMode = HY_PREPARE_TIMEOUT_MODE | HY_ACTION_TIMEOUT_MODE来设置

```
typedef NS_OPTIONS(int, HYShowTimeOutMode) {
    HYShowTimeOutMode_TIMEOUT_HIDDEN = 1 << 0, // 隐藏所有倒计时
    HYShowTimeOutMode_PREPARE = 1 << 1, // 准备阶段倒计时
    HYShowTimeOutMode_LONGCHECK = 1 << 2, // 耗时检测阶段倒计时
    HYShowTimeOutMode_ACTION = 1 << 3, // 动作阶段倒计时
};
```

swift端设置方式

```
config.showTimeOutMode = [HYShowTimeOutMode.PREPARE, HYShowTimeOutMode.LONGCHECK]
//or
config.showTimeOutMode = HYShowTimeOutMode.init(rawValue:
HYShowTimeOutMode.PREPARE.rawValue | HYShowTimeOutMode.LONGCHECK.rawValue)
// 全部隐藏
config.showTimeOutMode = [HYShowTimeOutMode.TIMEOUT_HIDDEN]
```


5. SDK接口使用说明

5.1 初始化配置，并拉取配置参数

在使用慧眼SDK之前，需要调用此方法传入基本配置参数，同时通过回调拉取本地的配置参数信息

```
// HuiYanOs的相关参数
HuiYanPrivateConfig *config = [[HuiYanPrivateConfig alloc] init];
// license文件在bundle中的路径
config.authLicense = [[NSBundle mainBundle] pathForResource:@"xx.license" ofType:@""];
// 活体检测本地检测超时时间 (ms)
config.authTimeOutMs = 20000;
//设置 HuiYanPrivateDelegate 代理
config.delegate = delegate;
// 启动核身前，拉取本地的配置参数信息
[HuiYanPrivateApi startGetAuthConfigData:config
withSuccCallback:^(PrivateGetConfigResult * _Nonnull getConfigResult) {
    // 获取配置信息成功，将配置信息发送给服务器，兑换启动核身配置，服务器下发的光线序列（客户自己实现）
    PrivateLiveDataEntity *liveDataEntity= [self getLightDataWith:getConfigResult];
    [self startAuthWithLiveData:liveDataEntity];
} withFailCallback:^(int errorCode, NSString * _Nonnull errMsg) {
    // 获取配置参数失败（客户自己实现）
    NSLog(@"errorCode:%d, errMsg:%@", errorCode, errMsg);
}];
```

5.2 完成剩余步骤活体核身

当您已经将配置信息从服务器端兑换完成之后，将服务器下发的LightData也就是核身的光线序列，通过此接口传入继续完成剩余本地核身功能。

```
[HuiYanPrivateApi startAuthByLiveData:liveDataEntity
withSuccCallback:^(PrivateCompareResult * _Nonnull comparaResult, NSString * _Nonnull
videoPath) {
    //活体通过检测结果数据
    // 1. 将本地核身的数据信息，发送到服务器端做比对验证，得到最终结果。（客户自己实现）
    [self checkAuthResultByData:comparaResult];
    // 2. 处理本地核身视频videoPath。（客户自己实现）
    [self dealWithAuthVideo:videoPath];
} withFailCallback:^(int errorCode, NSString * _Nonnull errMsg) {
    // 本地核身失败获取，发生错误
    NSLog(@"errorCode:%d, errMsg:%@", errorCode, errMsg);
}];
```

5.3 SDK资源释放

在您APP退出使用的时候，可以调用SDK资源释放接口

```
// 退出时做资源释放
- (void)dealloc {
    [HuiYanPrivateApi release];
}
```

5.4 常见问题与解决方法

1. 进入SDK，出现crash 打印日志 **"reason: '*** -[__NSDictionaryM setObject:forKey:]: key cannot be nil'"**,处理方法是在build Settings - Other Linker Flags 添加 **-ObjC**
2. 编译时报：
Undefined symbol: _vImageConvert_Planar16FtoPlanarF
Undefined symbol: _vImageConvert_PlanarFtoPlanar16F
处理方法：添加系统库Accelerate.framework
3. **"face-tracker-v003 bundle path is nil"** 或者 **HuiYanSDKUI bundle path is nil** 添加两个bundle 资源文件至Copy Bundle Resources
4. 自定义封装bundle静态资源库步骤可参考：<https://cloud.tencent.com/developer/article/1133094> Bundle 打包