

iOS安全SDK使用文档-Haotian

1. 背景

为了使企业移动端iOS应用能更便捷的整合端上安全能力，确保服务端接入安全业务的质量，企业需要集成iOS安全SDK。

此文档用于Haotian平台，v2.6.5.2版本，2023年3月28日发布。

2. 项目工程集成SDK方式

2.1 集成SDK

安全SDK以静态库的形式分发，解压后按照以下步骤集成到项目工程中：

1. 将libHTSSDKLib.a、HTSSDKLib.h添加进工程。
2. 设置【Other Linker Flags】选项为-ObjC标识
3. 在【Build Phases】的【Link Binary With Libraries】中添加系统库：
 - libz.tbd
 - libc++.tbd

注意事项

1. 由于Xcode14(2022年发布)已经默认不生成bitcode，所以iOS安全SDK不再包含bitcode，如有需求请用之前的SDK版本；
2. iOS安全SDK静态库只包含ARMv7/ARMv7s/ARM64架构，不支持模拟器x86架构。可在调用安全SDK接口前加入以下预编译宏，避免在模拟器x86架构中引入安全SDK：

</>

Object-C | 收起 ^

```
1 // Objective-C 使用
2 #if !(TARGET_OS_SIMULATOR)
3 // 调用安全SDK接口
4 #endif
5
6 // Swift 使用
7 #if !(arch(i386) || arch(x86_64))
8 // 调用安全SDK接口
9 #endif
```

3. 业务方调用安全SDK流程

3.1 申请产品信息

调用前，企业可通过百度智能云-昊天镜业务安全风控AFD控制台中，上传应用，申请AppKey和SecretKey，用于初始化接口。

</>

Plain Text | 收起 ^

```
1 - (void)startSDKEngineWithDeviceID:(NSString *)deviceID
2         appKey:(NSString *)appKey
3         secretKey:(NSString *)secretKey
4         ZInfoReadyHandler:(void (^)(NSError
    *error))ZInfoReadyHandler;
```

3.2 启动SDK

在AppDelegate文件中的didFinishLaunchingWithOptions方法中，尽可能靠前启动安全SDK，传入申请的产品信息，同时注册callback block。安全SDK初始化完成后，会调用callback block。

</>

Object-C | 收起 ^

```
1 - (BOOL)application:(UIApplication *)application
  didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
2     ...
3
4     HTSSDKLib *lib = [HTSSDKLib sharedInstance];
5     [lib startSDKEngineWithDeviceID:@"device_id_example"
6         appKey:@"app_key_example"
7         secretKey:@"secrete_key_example"
8         ZInfoReadyHandler:^(NSError *error) {
9
10    }];
11    ...
12
13    return YES;
14 }
```

3.3 获取ztoken

业务线App需要调用**获取ztoken**接口得到ztoken，但是建议优先调用startSDKEngineWithDeviceID接口进行SDK启动。

同时接口中需要传入EventID和AccountID。AccountID代表企业的用户ID，用于后续企业和服务打通使用。EventID请参考下面3.4。

注意事项

1. 确保**获取ztoken接口**调用之前，已经调用了**初始化安全SDK接口**；
2. ztoken具有时效性，业务方客户端每次上报前务必重新调用**获取ztoken接口**，获取当前最新的**ztoken**，业务方避免将ztoken写入Cookie；
3. 避免将ztoken发送到和使用ztoken无关的服务流量中。

3.4 确认EventID使用场景

EventID代表事件行为标识，用于调用**获取ztoken接口**获取ztoken信息时的必传参数，放在参数NSDictionary中，key为kSSDKEventIdentifier_ht，value为字符串3300~3400，根据业务可自行选择传入。

4. 使用SDK注意事项（非常重要）

为确保安全SDK相关接口使用正确，能够正确获取有效ztoken，以及提高后续进行风控接口查询结果的准确性，需要注意以下几点。

4.1 确保第一时间启动SDK

企业App集成安全SDK后，在启动过程中，需要将安全SDK的**启动接口调用**尽可能前置。

安全SDK启动接口被调用后，SDK的启动过程不会block当前线程，SDK启动完毕后通过callback回调通知业务逻辑初始化完毕，此时可以获取到当前最新的有效ztoken。

错误使用案例：

1. 启动安全SDK的逻辑依赖业务其他逻辑配置，由于业务逻辑配置（受环境或其他影响）无法正常运转导致部分场景没有启动安全SDK，大量设备获取到错误token；
2. 没有及时启动安全SDK，并且不等启动SDK的方法callback回调完成，就开始调用**获取ztoken接口**，此时可能大量获取到错误状态的ztoken

4.2 ztoken禁止缓存处理

因为ztoken有时效性，而且可能会因为客户端某些状态的变化实时发生变化，使用时必须确保“**每次发送给服务端前，立即调用一次接口获取最新ztoken，不要使用缓存的ztoken**”

4.3 ztoken使用场景

只有当企业App进行**关键活动和行为**时才获取ztoken，并且只有**关键活动和行为**触发的客户端请求才能携带ztoken，业务方服务端在收到ztoken后，用此ztoken进行风控查询。

注意事项

- 1. 业务方避免将ztoken写入Cookie；
- 2. 避免将ztoken发送到和使用ztoken无关的服务流量中。

5. 主要接口说明

5.1 获取SDK实例对象接口

接口描述

</>Object-C | 收起 ^

```
1 + (id)sharedInstance;
```

用于获取安全SDK的单一实例对象，后续接口通过该实例对象调用。

参数说明

无

返回值说明

安全SDK单一实例对象

5.2 初始化安全SDK接口

安全SDK进行初始化时，请直接调用此接口。

接口描述：

</>Object-C | 收起 ^

```
1 - (void)startSDKEngineWithDeviceID:(NSString *)deviceID
2                               appKey:(NSString *)appKey
3                               secretKey:(NSString *)secKey
4                               ZInfoReadyHandler:(void (^)(NSError
    *error))ZInfoReadyHandler;
```

参数说明

- 1. deviceID: 业务方使用的唯一设备标识（业务方对该设备的描述）；
- 2. appKey: 宿主App申请的5位appKey；
- 3. secretKey: 宿主App申请的32位secretKey；

4. ZInfoReadyHandler: 业务方为安全SDK提供的回调函数，该函数被调用时，如果error为nil则意味着[获取ztoken接口](#)可以正确返回。

关于回调函数 ZInfoReadyHandler

初始化安全SDK接口为非阻塞操作，调用后会立即返回；安全SDK通过异步回调的机制（ZInfoReadyHandler）通知业务方。

注意事项

- 1. appKey与secretKey需要在接入前向安全SDK申请，否则相关服务不可用；
- 2. 回调函数只会至多被调用一次；
- 3. 为确保打通服务的ztoken质量，安全SDK会在状态就绪后第一时间回调ZInfoReadyHandler；网络异常等特殊情况下，该回调函数会被延迟调起。

5.3 获取ztoken接口

业务方Native需要获取ztoken接口时使用。

接口描述

</> Object-C | 收起 ^

```
1 - (NSDictionary *)getZInfoWithEvent:(NSDictionary *)eventInfo
```

在业务方特定事件 / 场景（如支付、登录、参加运营活动等事件）发生时，通过此接口获取当前有效ztoken，立即上传至服务端，服务端利用ztoken接入安全SDK服务对该设备风险进行评估。

参数说明

eventInfo字典用于区分业务方不同的事件或场景，“Key-Value”遵循以下格式：

1	Key	Value Type	示例	说明
2	kSSDKEventIdentifier	NSString*	@"3348"	预先申请的eventID，对应当前获取ztoken的场景
3	kSSDKAccountIdentifier	NSString*	@"10538532"	当前登录用户的唯一标识

示例代码如下：

</> Object-C | 收起 ^

```
1 - (void)onUserPaymentEvent() { // 特定事件触发
2     /* 准备参数 */
```

```

3   NSString *eventID = @"3348";           //事先申请的事件标识字符串
4   NSString *accountID = @"10538532";     //当前用户的唯一标识
5   NSDictionary *eventInfo = @{
6       kSSDKEventIdentifier_ht: eventID,
7       kSSDKAccountIdentifier_ht: accountID
8   };
9
10  /* 获取ztoken */
11  HTSSDKLib *sdk = [HTSSDKLib sharedInstance];
12  NSDictionary *ZInfo = [sdk getZInfoWithEvent:eventInfo];
13  NSString *ztoken = [ZInfo valueForKey:kSSDKZToken_ht];
14
15  /*
16   * 正常情况下ztoken为BASE64字符串，例如：
17   *
18   * uSLPqWj2s-8QZ3auiig1MDK4pdPdg5YAkVgdc1fynhUqAFGtV7h1XfKeZ2uZswTeXsqQqv4dErXyv
19   * CpbckU8ufw
20   */
21  ...
22  }

```

返回值说明

调用返回ZInfo字典，ZInfo中kSSDKZToken_ht对应的value为ztoken。ZInfo的格式如下图所示，可能的返回值见示例代码。

1	Key	Value Type	示例	说明
2	kSSDKZToken_ht	NSString*	见示例代码	ztoken