



## 快速集成

文档版本: 20240826

[查看在线版本](#)

## 目录

<b>1. SDK 版本 .....</b>	<b>2</b>
<b>2. 集成 SDK .....</b>	<b>3</b>
2.1. 构建和下载 SDK .....	3
2.2. 使用 CocoaPods 集成 .....	4
2.3. 初始化 SDK .....	4
2.4. 配置 ThingSmartHomeKit .....	5
2.5. 配置 ThingSmartBusinessExtensionKit .....	6
2.6. 打开 Debug 模式 .....	6
2.7. 语言设置 .....	7
2.8. 配置 Widget 工程 .....	7
<b>3. Demo App .....</b>	<b>9</b>
3.1. 准备 Demo .....	9
3.2. Demo 概述 .....	9
<b>4. 常见问题 .....</b>	<b>11</b>
4.1. SING_VALIDATE_FALED .....	11

---

本文介绍如何使用 CocoaPods 将涂鸦智能生活 App SDK 快速集成到您的项目中。SDK 要求 iOS 版本不低于 9.0。

## 1. SDK 版本

- 如果您采用的 5.x.x 以下版本 SDK，请提前根据 [迁移指南](#) 进行升级。
  - 若您已经集成过低版本 SDK，更新至 5.0 SDK 后，请删除项目中的低版 `t_s.bmp` 安全图片文件，并从 [涂鸦开发者平台](#) 上获取新的 5.0 版本 App 密钥。
  - 从 4.0.0 版本智能生活 App SDK 开始，SDK 分为开发版和正式版。详情请参考 [产品定价](#)。  
**开发版 SDK 仅供开发使用，请勿直接商用。如需上架商用，请前往平台购买正式版。购买正式版 SDK 后：**
- 需在涂鸦开发者平台重新构建正式版 SDK 并下载。
  - 重新在项目中集成正式版 SDK。

## 2. 集成 SDK

### 2.1. 构建和下载 SDK

- 前往 [涂鸦开发者平台](#)。
- 选择 5.x.x 版本 SDK 或其他 UI 业务包。

**i**

若之前使用过低版本 SDK，右上角可切换版本。

The screenshot shows the 'device test' configuration page. At the top, it displays 'device test' and '修改 App 名称'. Below that, it shows 'iOS 包名: con' and 'Android 包名: co.' with a placeholder '渠道标识符: d'. There are three tabs at the top: '获取SDK' (selected), '获取密钥', and '独立域名服务'. In the bottom right corner of the main area, there is a red-bordered button labeled '切换至旧版本: V4'.

**基础配置**

**智能生活 APP SDK** (必选)  
5.0.1  
是一套针对全屋智能家居领域的移动端软件开发包。集成后，开发者可以快速的实现自有 App 对智能设备的连接与控制。

**垂直类产品功能 SDK**

- 出行 SDK**  
封装了智慧出行设备相关的功能接口，加速开发过程。
- IPC SDK**  
是一套针对网络摄像场景下的移动端软件开发包。
- 扫地机 SDK**  
是一套针对扫地机类智能设备的移动端软件开发包。

**UI 业务包**

- 设备控制 UI 业务包**  
它提供了设备控制面板的加载和控制业务逻辑和 UI 封装，加速应用开发过程。
- IPC UI 业务包**  
它提供了摄像机功能相关的面板业务逻辑和 UI 封装，加速应用开发过程。
- 商城 UI 业务包**  
它提供了强大的商城业务逻辑和 UI 封装，加速应用开发过程。

- 选择构建 SDK。

The screenshot shows the 'device test' configuration page with the '获取SDK' tab selected. It displays the same basic configuration options as the previous screenshot. In the bottom right corner, there is a red-bordered button labeled '构建 iOS 开发版'.

**基础配置**

**智能生活 APP SDK** (必选)  
5.0.1  
是一套针对全屋智能家居领域的移动端软件开发包。集成后，开发者可以快速的实现自有 App 对智能设备的连接与控制。

**垂直类产品功能 SDK**

- 出行 SDK**  
封装了智慧出行设备相关的功能接口，加速开发过程。
- IPC SDK**  
是一套针对网络摄像场景下的移动端软件开发包。
- 扫地机 SDK**  
是一套针对扫地机类智能设备的移动端软件开发包。

**UI 业务包**

- 设备控制 UI 业务包**  
它提供了设备控制面板的加载和控制业务逻辑和 UI 封装，加速应用开发过程。
- 常见问题与反馈 UI 业务包**  
它提供了常见问题排查与意见反馈渠道的业务逻辑和 UI 封装，加速应用开发过程。
- 设备配网 UI 业务包**  
它提供了多种智能设备的配网初置操作引导和入网激活的业务逻辑和 UI 封装，加速应用开发过程。
- 设备详情 UI 业务包**  
它提供了设备信息编辑、查询、备用网络设置等业务逻辑和 UI 封装，可以帮助你快速实现与设备详情相关的能力。
- IPC UI 业务包**  
它提供了摄像机功能相关的面板业务逻辑和 UI 封装，加速应用开发过程。
- 消息中心 UI 业务包**  
它提供了消息中心业务逻辑和 UI 封装，加速应用开发过程。
- OTA UI 业务包**  
它提供了设备 OTA 固件升级业务逻辑和 UI 封装，可以帮助你快速集成设备固件升级的能力。
- 云端 UI 业务包**  
为 C 端用户提供开通 IPC 产品视频云存储服务能力。
- 家庭管理 UI 业务包**  
它提供了家庭管理、成员管理、房间管理等业务逻辑和 UI 封装。
- 群组管理 UI 业务包**  
支持创建、编辑与控制 WiFi 智能设备群组。

在底部，有三个按钮：'构建 iOS 开发版' (red border), '构建 Android 开发版', '快速集成 iOS SDK', '快速集成 Android SDK'.

- 构建完成后，下载 SDK 进行集成。

[下载 iOS 开发版](#)[构建 Android 开发版](#)

5. 解压 `ios_core_sdk.tar.gz`，得到以下两个重要文件信息：

- Build：存放您的**App 专属安全 SDK**，它和 App 密钥信息一样重要，请妥善保管，**谨防资源泄漏或对外公开。**
  - `ThingSmartCryption.podspec`：用于在引用、集成使用 5.0 SDK。
6. (建议) 将以上两个文件与 `podfile` 保持同级存放，便于后续引用操作。
7. `ThingSmartBusinessExtensionKit` 是 `ThingSmartHomeKit` 的高级封装组件，不仅包含了 `ThingSmartHomeKit` 的所有功能，还提供了更多便捷的扩展能力。因此，**我们更推荐直接使用 `ThingSmartBusinessExtensionKit`**。

## 2.2. 使用 CocoaPods 集成

1. 更新 CocoaPods 至最新版本。CocoaPods 的使用请参考 [CocoaPods 官方文档](#)。
2. 在 `Podfile` 文件中添加以下内容：

```
1 source 'https://github.com/tuya/tuya-pod-specs.git'
2 platform :ios, '11.0'
3 target 'Your_Project_Name' do
4     # 从涂鸦开发者平台 (platform.tuya.com) 构建和获取 ThingSmartCryption
5     # 购买正式版后，需重新在涂鸦开发者平台构建 SDK 并重新集成
6     # ./ 代表将 ios_core_sdk.tar.gz 解压之后所在目录与 podfile 同级
7     # 若自定义存放目录，可以修改 path 为自定义目录层级
8     pod "ThingSmartCryption", :path => './'
9     pod "ThingSmartHomeKit"
10
11    # ThingSmartBusinessExtensionKit 不仅包含了 ThingSmartHomeKit 的所有功能，还
12    提供了更多便捷的扩展能力。
13    pod "ThingSmartBusinessExtensionKit"
14 end
15 target 'Your_Extension_Target_Name' do
16    # 所有扩展程序 target，只要引入 ThingSmartHomeKit 则必须同时引入 ThingSmartCryption
17    on
18        pod "ThingSmartCryption", :path => './'
19        pod "ThingSmartHomeKit"
20    # ThingSmartBusinessExtensionKit 不仅包含了 ThingSmartHomeKit 的所有功能，还
21    提供了更多便捷的扩展能力。
22    pod "ThingSmartBusinessExtensionKit"
23 end
```

3. 如果您的 App 包含扩展程序 `target`，例如 Siri 和 Widget，确保 `Podfile` 配置正确。
4. 在项目根目录下，执行 `pod update` 命令进行集成。

## 2.3. 初始化 SDK

1. 在 **Target > General** 中打开项目设置，修改 `Bundle Identifier` 为您在 [涂鸦开发者平台](#) 设置的 iOS 包名。
2. 在项目的 `PrefixHeader.pch` 文件添加以下内容：

```
1 #import <ThingSmartHomeKit/ThingSmartKit.h>
```

如果是 Swift 项目，可以在 `xxx_Bridging-Header.h` 桥接文件中添加以下内容：

```
1 #import <ThingSmartHomeKit/ThingSmartKit.h>
```

3. 若集成了 `ThingSmartBusinessExtensionKit` 组件，可以引入以下头文件：

```
1 #import <ThingSmartBusinessExtensionKit/ThingSmartBusinessExtensionKit.h>
```

如果是 Swift 项目，可以在 `xxx_Bridging-Header.h` 桥接文件中添加以下内容：

```
1 #import <ThingSmartBusinessExtensionKit/ThingSmartBusinessExtensionKit.h>
```

1. 打开 `AppDelegate.m` 文件，在 `[AppDelegate application:didFinishLaunchingWithOptions:]` 方法中初始化 SDK。

## 2.4. 配置 ThingSmartHomeKit

```
1 - (void)startWithAppKey:(NSString *)appKey secretKey:(NSString *)secretKey;
```

### 参数说明

参数	说明	来源
appKey	App 唯一凭证信息	访问 <a href="#">涂鸦开发者平台的 SDK</a> 详情页，选择目标 App，并 单击 <a href="#">获取密钥</a> 页签
secretKey	App 密钥 key	访问 <a href="#">涂鸦开发者平台的 SDK</a> 详情页，选择目标 App，并 单击 <a href="#">获取密钥</a> 页签

参数	说明	来源
bundleId	iOS App 应用包名	访问 <a href="#">涂鸦开发者平台的 SDK</a> <a href="#">详情页</a> ，选择目标 App，并找到 iOS 应用包名

### 示例代码

Objective C:

```
1 [[ThingSmartSDK sharedInstance] startWithAppKey:<#your_app_key#> sec  
2 retKey:<#your_secret_key#>];
```

Swift:

```
1 ThingSmartSDK.sharedInstance()?.start(withAppKey: <#your_app_key#>,  
2 secretKey: <#your_secret_key#>)
```

## 2.5. 配置 ThingSmartBusinessExtensionKit

在 App 启动后调用 `loadConfig` 配置业务拓展 SDK。

```
1 - (BOOL)application:(UIApplication *)application didFinishLaunch  
2 ingWithOptions:(NSDictionary *)launchOptions  
3 {  
4     [ThingSmartBusinessExtensionConfig setupConfig];  
5     return YES;  
6 }
```

至此，准备工作已经全部完毕。了解 `ThingSmartBusinessExtensionKit` 入门请参考 [业务拓展 SDK 开发教程](#)。

## 2.6. 打开 Debug 模式

在开发的过程中可以开启 Debug 模式，打印日志用于分析问题。

Objc:

```
1 #ifdef DEBUG  
2     [[ThingSmartSDK sharedInstance] setDebugMode:YES];  
3 #else  
4 #endif
```

Swift:

```
1 #if DEBUG
2     ThingSmartSDK.sharedInstance()?.debugMode = true
3 #else
4 #endif
```

## 2.7. 语言设置

接口返回的错误信息与文案等，将会根据您工程所支持的多语言和用户手机的当前语言来显示。如果想要支持某种语言，可以在工程中的 **Localization** 中添加您想要支持的语言。

演示工程主要介绍了 SDK 开发流程，以及一些简单的业务逻辑。在开发 App 之前，建议先按照本文完成演示工程的操作。

## 2.8. 配置 Widget 工程

Widget 创建步骤如下：

1. 修改 Podfile。

```
1 target 'Your_Extension_Target_Name' do
2     # 1. 所有扩展程序 target，只要引入 ThingSmartHomeKit 则必须同时引入 ThingSmartCry
3     ption
4     # 2. 从涂鸦开发者平台 (platform.tuya.com) 构建和获取 ThingSmartCryption
5     # 购买正式版后，需重新在涂鸦开发者平台构建 SDK 并重新集成
6     # ./ 代表将 ios_core_sdk.tar.gz 解压之后所在目录与 podfile 同级
7     # 若自定义存放目录，可以修改 path 为自定义目录层级
8     pod "ThingSmartCryption", :path => './'
9     pod "ThingSmartHomeKit"
10 end
11 post_install do |installer|
12     installer.pods_project.targets.each do |target|
13         target.build_configurations.each do |config|
14             config.build_settings['APPLICATION_EXTENSION_API_ONLY'] =
15 = 'NO'
16         end
17     end
18 end
```

- 配置 Target，在 Target 中按需引入组件或 SDK。
  - 需要注意，引入 SDK 的同时必须引入 ThingSmartCryption。
2. 设置 AppGroups。
    - 开启 AppGroups 权限。
    - 在 SDK 初始化之前，给 SDK 设置 App Groups Name。

- AppGroups 只有付费开发者账号才能创建权限，所以免费开发者账号不能调试 Widget 应用。

3. 使用 AppKey 和 AppSecret 初始化 SDK。
4. 在用 AppKey 初始化 SDK 之前，设置 App Groups Name。

```
1 [ThingSmartSDK sharedInstance].appGroupId = APP_GROUP_NAME;
2 [[ThingSmartSDK sharedInstance] startWithAppKey:SDK_APPKEY secretKey
3 :SDK_APPSECRET];
```

## 3. Demo App



智能生活 App SDK 示例工程 Demo 仅用于演示体验，请勿直接商用。更多详情，请访问[《涂鸦开发服务协议》](#)。

### 3.1. 准备 Demo

在[准备工作](#)阶段，您可以获取 iOS 版本 SDK 的 AppKey 和 AppSecret。

集成 SDK 时，请确认 BundleId、AppKey 和 AppSecret 是否与涂鸦开发者平台上的信息一致，任意一个不匹配会导致 SDK 无法使用。

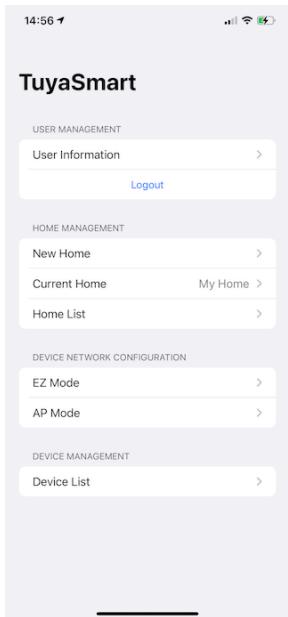


从 5.0 SDK 切换到其他 SDK 版本后，需重替换为对应版本的密钥信息和更换集成方式

### 3.2. Demo 概述

演示工程提供了 Swift 与 Objective-C 的版本，请访问[Swift 示例](#)与[Objective-C 示例](#)获取演示工程。工程主要包括了以下功能：

- 用户账号注册及登录
- 用户信息管理
- 家庭管理
- Wi-Fi 快连和热点模式配网
- 设备控制



## 4. 常见问题

### 4.1. SING\_VALIDATE\_FALED

- **问题现象：**API 请求提示以下签名错误。

```
1  {
2      "success" : false,
3      "errorCode" : "SING_VALIDATE_FALED",
4      "status" : "error",
5      "errorMsg" : "Permission Verification Failed",
6      "t" : 1583208740059
7 }
```

- **解决办法：**确认 BundleId、AppKey 和 AppSecret 是否与 [涂鸦开发者平台](#) 上的信息一致，任意一个不匹配都将校验失败。具体请按照 [准备工作](#) 来进行检查。
- **注意事项：**如果购买正式版，需重新在涂鸦开发者平台构建 SDK 并替换密钥。