Fast Integration with Smart Life App SDK for iOS

Version: 20240826

Online Version

Contents

1. SDK versions 2

2. Integrate with the SDK 3

2.1. Build and download the SDK 3
2.2. Use CocoaPods for fast integration 4
2.3. Initialize the SDK 5
2.4. Configure the ThingSmartHomeKit 6
2.5. Configure ThingSmartBusinessExtensionKit 7
2.6. Enable the debug mode 7
2.7. Configure multilingual options 8
2.8. Create a widget 8

3. Demo app 10

3.1. Prepare the demo 10
3.2. Demo features 10

4. FAQs 12

4.1. SING_VALIDATE_FALED 12

This topic describes how to use CocoaPods to quickly integrate Tuya Smart Life App SDK for iOS into your project. The SDK requires iOS 9.0 or later.

1. SDK versions

- If you have integrated Smart Life App SDK versions earlier than v5.x.x into your project, follow the instructions in Upgrade Guide and upgrade to the latest version.
- If you have integrated legacy SDK versions into your project, after you upgrade to v5.0, delete the legacy security image file `t_s.bmp` from your project, and get the app key information for v5.0 from the Tuya Developer Platform .
- Starting from Smart Life App SDK for Android v4.0.0, the SDK is classified into the development edition and official edition. For more information, see Pricing . **The development edition is suitable for personal non-commercial scenarios only. Do not use it for commercial purposes. If your app is planned to be launched on app stores or in other commercial scenarios, go to the Tuya Developer Platform and purchase the official edition**. After the official edition is purchased:

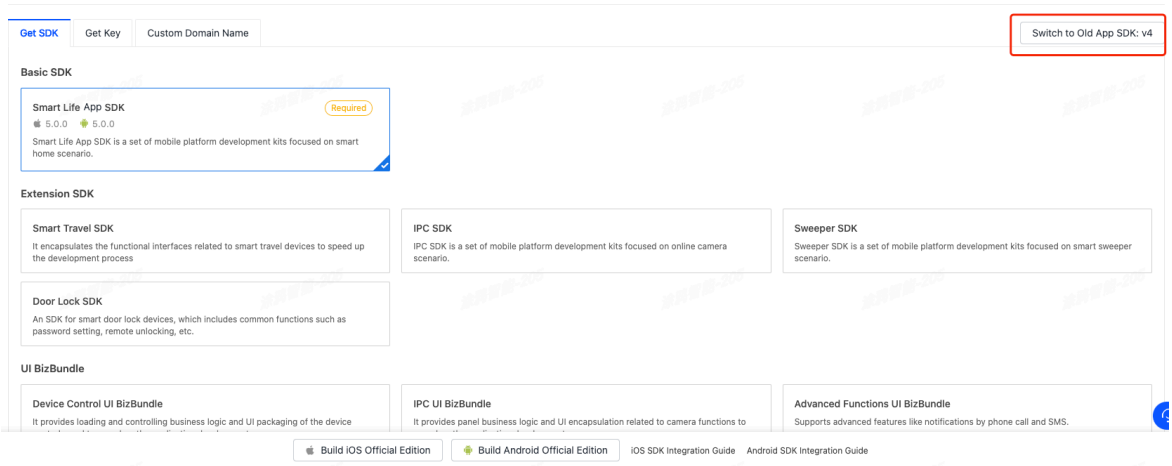
1. Rebuild the SDK and download the package of the official edition on the platform.
2. Integrate the SDK of the official edition into your project.

2. Integrate with the SDK

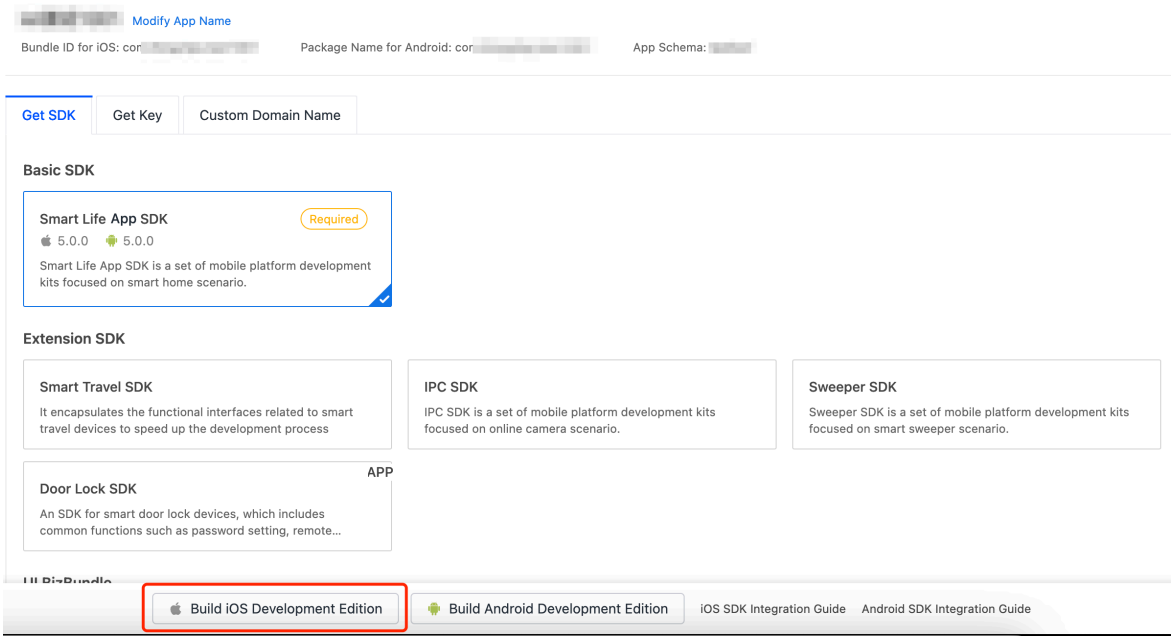
2.1. Build and download the SDK

1. Log in to the Tuya Developer Platform .
2. Select the required SDKs or UI BizBundles of v5.x.x.

> *i*
>
> If a legacy SDK version has been used, you can click the button in the top right corner to switch between the legacy and new versions.



3. Select the required SDKs or BizBundles and build your SDK.



4. After the build is finished, download the SDK to be integrated.

5. Extract `ios_core_sdk.tar.gz` and get the following important files:
- `Build` : stores the **security SDK exclusive to your app**. This file is as important as the app key information. Keep the file properly and **do not disclose the information in it**.
- `ThingSmartCryption.podspec` : used to reference and integrate with App SDK v5.0.
6. We recommend that you store both files at a sibling directory as `podfile` , so they can be referenced easily during subsequent development.
7. `ThingSmartBusinessExtensionKit` is an advanced encapsulation of `ThingSmartHomeKit` . It not only includes all the features of `ThingSmartHomeKit` , but also offers additional convenient capabilities. Therefore, **we highly recommend using `ThingSmartBusinessExtensionKit` .**

2.2. Use CocoaPods for fast integration

1. Update CocoaPods to the latest version. For more information about CocoaPods, see CocoaPods Guides .
2. Add the following code block to the `Podfile` :

```
source 'https://github.com/tuya/tuya-pod-specs.git'
platform :ios, '11.0'
target 'Your_Project_Name' do
    # Build and get ThingSmartCryption from the Tuya Developer Platf
orm (platform.tuya.com).
    # After the official edition is purchased, rebuild the SDK on th
e Tuya Developer Platform and integrate it into your project.
    # The dot slash (./) notation represents that the files that are
 obtained after ios_core_sdk.tar.gz is extracted are put at a siblin
g directory as podfile.
    # To use another directory, change the path to your desired dire
ctory.
    pod "ThingSmartCryption", :path =>'./'
    pod "ThingSmartHomeKit"
    # ThingSmartBusinessExtensionKit not only includes all the featu
res of ThingSmartHomeKit, but also offers additional convenient capa
bilities.
    pod "ThingSmartBusinessExtensionKit"
end
target 'Your_Extension_Target_Name' do
    # Regarding all extension targets, as long as ThingSmartHomeKit
is imported, ThingSmartCryption must be imported too.
    pod "ThingSmartCryption", :path =>'./'
```

```
    pod "ThingSmartHomeKit"
    # ThingSmartBusinessExtensionKit not only includes all the featu
res of ThingSmartHomeKit, but also offers additional convenient capa
bilities.
    pod "ThingSmartBusinessExtensionKit"
end
```

3. If your app contains extension `targets`, such as Siri and Widgets, make sure the `Podfile` is configured correctly.
4. In the root directory of your project, run `pod update`.

2.3. Initialize the SDK

1. Choose **Target** > **General** to open the project settings, and modify `Bundle Identifier` to the iOS Bundle ID of the app that is registered on the Tuya Developer Platform.
2. Add the following content to the `PrefixHeader.pch` file:

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

Add the following content to the bridging header file `xxx_Bridging-Header.h` for a Swift project:

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

3. If you have integrated the `ThingSmartBusinessExtensionKit` component, you can import the following header file:

```
#import <ThingSmartBusinessExtensionKit/ThingSmartBusinessExtensionK
it.h>
```

For Swift projects, you can add the following to your `xxx_Bridging-Header.h` bridging header file:

```
#import <ThingSmartBusinessExtensionKit/ThingSmartBusinessExtensionK
it.h>
```

4. Open the `AppDelegate.m` file and initialize the SDK in `[AppDelegate application:didFinishLaunchingWithOptions:]` .

2.4. Configure the ThingSmartHomeKit

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

Parameters

Parameter	Description	Source
appKey	The credential information exclusive to the app.	Go to Tuya Developer Platform > Details page of your SDK-based app , select the desired app, and then click the **Get Key** tab
secretKey	The secret key of the app.	Go to Tuya Developer Platform > Details page of your SDK-based app , select the desired app, and then click the **Get Key** tab
bundleId	The Bundle ID for iOS.	Go to Tuya Developer Platform > Details page of your SDK-based app , select the desired app, and then find Bundle ID for iOS

Sample code

Objective-C:

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

Swift:

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

2.5. Configure ThingSmartBusinessExtensionKit

After the app launches, call `loadConfig` to configure the BizBundle SDK.

```
    - (BOOL)application:(UIApplication *)application didFinishLaunch
ingWithOptions:(NSDictionary *)launchOptions
    {
        [ThingSmartBusinessExtensionConfig setupConfig];
        return YES;
    }
```

Now, you are ready for app development. To learn how to get started with `ThingSmartBusinessExtensionKit` , refer to the BizBundle SDK Development Tutorial .

2.6. Enable the debug mode

During the development, you can enable the debug mode and print logs for troubleshooting.

Objective-C:

```
#ifdef DEBUG
    [[ThingSmartSDK sharedInstance] setDebugMode:YES];
#else
#endif
```

Swift:

```
#if DEBUG
   ThingSmartSDK.sharedInstance()?.debugMode = true
#else
#endif
```

2.7. Configure multilingual options

The returned error messages and other UI text are displayed in languages as configured in the multilingual settings of your project and users' mobile phone system languages. To support a certain language, add it to **Localization** of your project.

In the following example, a demo app is used to describe the process of app development with the App SDK. Before the development of your app, we recommend that you run the demo app.

2.8. Create a widget

Perform the following steps:

1. Modify the `Podfile` .

```
target 'Your_Extension_Target_Name' do
    # 1. Regarding all extension targets, as long as ThingSmartHomeK
it is imported, ThingSmartCryption must be imported too.
    # 2. Build and get ThingSmartCryption from the Tuya Developer Pl
atform (platform.tuya.com).
    # After the official edition is purchased, rebuild the SDK on th
e Tuya Developer Platform and integrate it into your project.
    # The dot slash (./) notation represents that the files that are
 obtained after ios_core_sdk.tar.gz is extracted are put in a siblin
g directory as podfile.
    # To use another directory, change the path to your desired dire
ctory.
    pod "ThingSmartCryption", :path =>'./'
    pod "ThingSmartHomeKit"
end
post_install do | installer |
     installer.pods_project.targets.each do | target |
        target.build_configurations.each do | config |
            config.build_settings['APPLICATION_EXTENSION_API_ONLY']
= 'NO'
        end
    end
end
```

- Configure `Target` and import components or SDKs into `Target` as needed.
- Note that `ThingSmartCryption` must be imported together with the SDK.

2. Configure `AppGroups` .

- Grant permissions on `AppGroups` .
- Before the SDK is initialized, set the `App Groups Name` for the SDK.

- Only paid developer accounts can be granted permissions on `AppGroups`. Therefore, free developer accounts cannot be used to debug widget applications.

3. Configure the `AppKey` and `AppSecret` to initialize the SDK.

1. Before the SDK is initialized with `AppKey`, set the `App Groups Name`.

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

3. Demo app

The demo app that is created in the sample project of Smart Life App SDK is used for reference only. Do not use the demo app for commercial purposes. For more information, see Tuya Development Service Agreement .

3.1. Prepare the demo

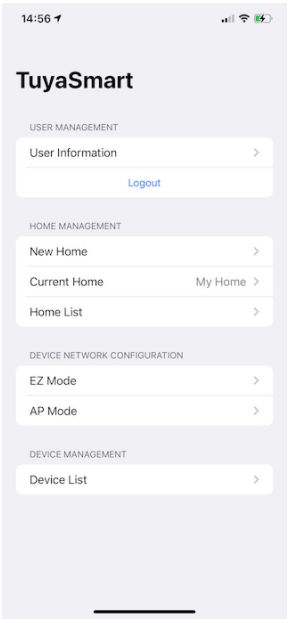
In the Preparation topic, get the `AppKey` and `AppSecret` for iOS.
Make sure that `BundleId` , `AppKey` , and `AppSecret` are consistent with those used on the Tuya Developer Platform. Any mismatch will cause the SDK development app to fail.

If the SDK is switched from v5.0 to another version, you must change the key information to that of the target version. The way the SDK is integrated is changed accordingly.

3.2. Demo features

The demo app is coded in Swift and Objective-C. You can get the sample in Swift and sample in Objective-C on GitHub. The following features are supported:

- User registration and login
- User management
- Home management
- Wi-Fi Easy Connect (EZ) mode and access point (AP) mode
- Device Control


14:56
TuyaSmart
USER MANAGEMENT
User Information
Logout
HOME MANAGEMENT
New Home
Current Home
My Home
Home List
DEVICE NETWORK CONFIGURATION
EZ Mode
AP Mode
DEVICE MANAGEMENT
Device List

4. FAQs

4.1. SING_VALIDATE_FALED

- **Problem**: When an API request is made, an error message is returned in the following response:

```
{
  "success" : false,
  "errorCode" : "SING_VALIDATE_FALED",
  "status" : "error",
  "errorMsg" : "Permission Verification Failed",
  "t" : 1583208740059
}
```

- **Solution**: Make sure that BundleId, AppKey, and AppSecret are consistent with those used on the Tuya Developer Platform . Any mismatch will cause the authentication to fail. For more information, see Preparation .
- **Note**: After you purchase the official edition, rebuild the SDK and replace the key information with the new one on the Tuya Developer Platform.