

# TingYunApp\_iOS\_SDK\_嵌码说明-V3.0

## 1、获取 TingYun SDK

### • CocoaPods集成

在项目工程的Podfile文件中添加以下代码：

```
pod 'tingyunApp'
```

若对数据传输有加密（国密算法加密）需求，可在项目工程的 Podfile 文件中添加以下代码：

```
pod 'NBSGKit'
```

若要使用「日志回捞」功能，可在项目工程的Podfile文件中添加以下代码：

```
pod 'TingyunLog'
```

若要使用「OOM崩溃采集」功能，可在项目工程的Podfile文件中添加以下代码：

```
pod 'NBSOOM'
```

若要使用「视频回放」功能，可在项目工程的Podfile文件中添加以下代码：

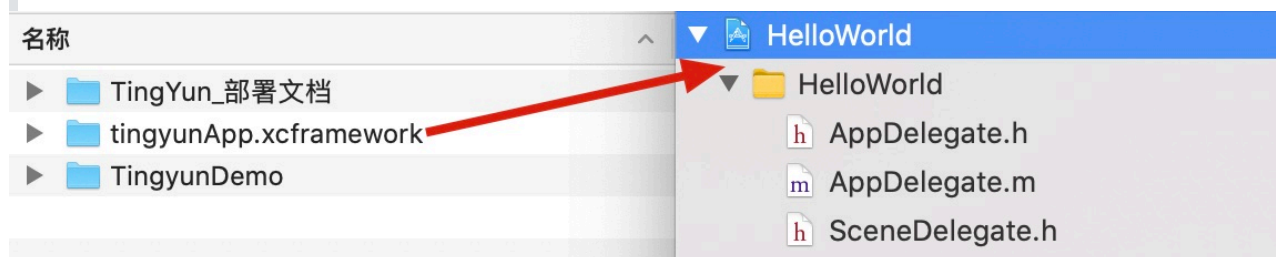
```
pod 'NBSReplayKit'
```

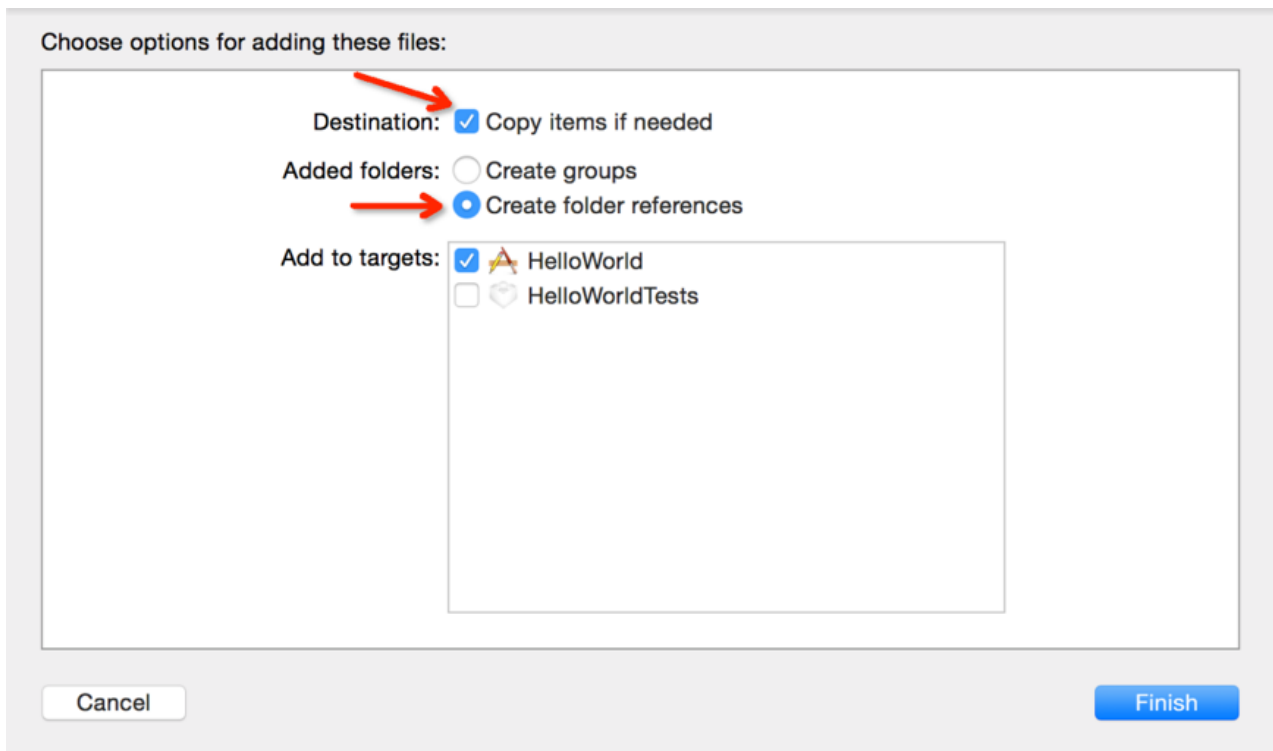
保存后执行 `pod install`

### • 手动集成

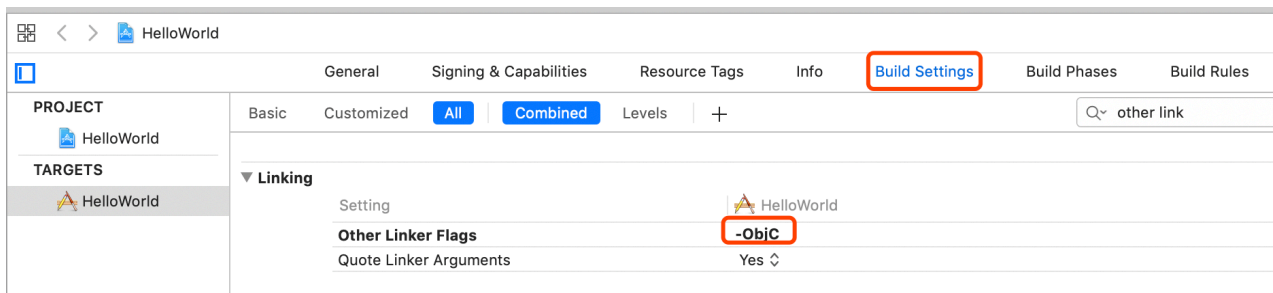
手动集成安装包请联系技术支持获取

#### 1.1、解压后将tingyunApp.xcframework拖拽到xcode工程中并勾选「Copy items if needed」

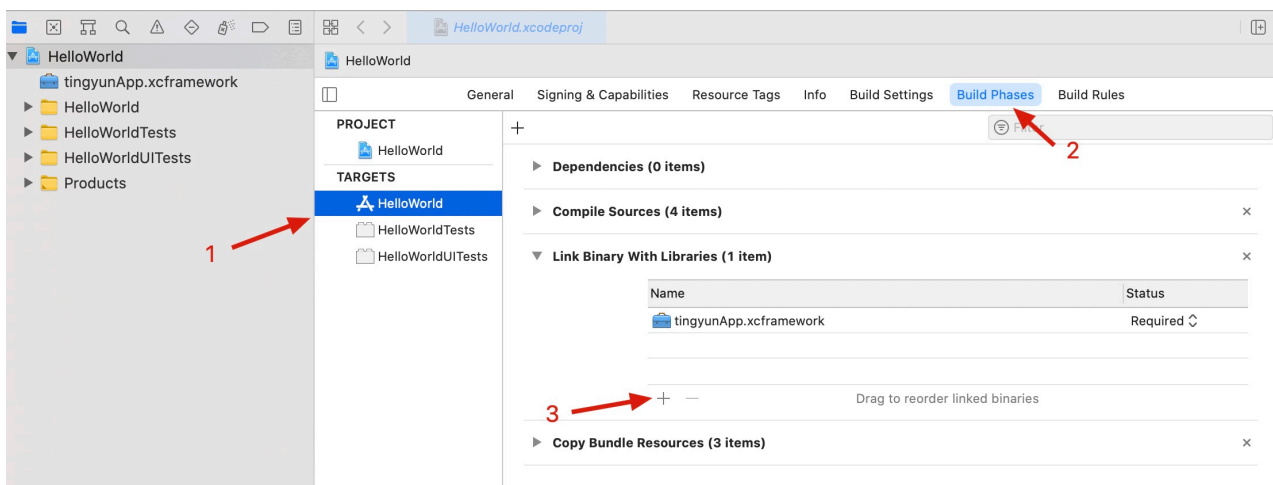




## 1.2、选择工程 TARGETS → Build Settings → 搜索「Other Linker Flags」并添加编译标志【-ObjC】



## 1.3、添加依赖库



依赖库	作用
libz.tbd	解压缩
WebKit.framework	浏览器内核
Security.framework	访问钥匙链
CoreTelephony.framework	获取运营商
SystemConfiguration.framework	获取网络状态
JavaScriptCore.framework	支持js自定义错误功能
CoreGraphics.framework	支持可视化命名功能
QuartzCore.framework	支持可视化命名功能
CFNetwork.framework	支持移动拨测功能
libresolv.tbd	支持移动拨测功能
libc++.tbd	c++符号化

若要SDK传输数据使用国密加密，请导入「NBSGKit.framework」并在工程 TARGETS → Genral → Framework,Libraries,and Embedded Content 中将「NBSGKit.framework」的「Embed」改为「Embed & Sign」

若要使用「日志回捞」功能，请导入「TingyunLog.xcframework」

若要使用「OOM崩溃采集」功能，请导入「NBSOOM.xcframework」

若要使用「视频回放」功能，请导入「NBSReplayKit.xcframework」

## 2、初始化SDK

### 2.1、获取基调听云host与AppKey

「App Key」及「host」地址请从基调听云平台中获取，详情参照下图

- 进入应用「设置」

搜索应用: <input type="text" value="请输入应用名称"/>											添加应用
应用名称	用户体验评分	活跃设备数	启动时间(ms)	可交互时间(ms)	首屏时间(ms)	操作时间(ms)	请求响应时间(ms)	请求错误率(%)	崩溃率(%)	卡顿率(%)	设置
应用名称	98	365	950	155	437	24	583	0.62	0.31	0	设置 启用 删除

- 在「基础设置」中复制「App Key」及「host」

## 应用设置

基础设置

安全设置

阈值设置

评分设置

请求设置

\* App名称: 演示账号

使用字母数字和中文，最长32个字符。

App Key: eb-13c21c23af4d3b616c8230c95c-1761

host: v.bridging.fun

## 2.2、导入头文件

- Objective-C 请在工程的「pch」文件中导入头文件

```
#import <tingyunApp/NBSAppAgent.h>
```

- Swift 请在工程的「Bridging\_Header.h」文件中导入头文件

```
#import <tingyunApp/NBSAppAgent.h>
```

注：若使用「日志回捞」功能，需导入日志库头文件：`#import <TingyunLog/TingyunLog.h>`

## 2.3、初始化

- Objective-C

在工程「main.m」文件中的「main」方法中初始化 SDK

若在AppDelegate.m中初始化SDK，则无法采集应用启动数据

```
int main(int argc, char * argv[]) {
    @autoreleasepool {
        //默认为https，如果redirect_host为http协议的话则需要调用该接口。
        [NBSAppAgent setHttpEnabled:YES];
        //redirect_host为平台服务器地址
        [NBSAppAgent setRedirectURL:@"redirect_host"];
        //Your_appkey从平台中获取
        [NBSAppAgent startWithAppID:@"Your_appkey"];
        return UIApplicationMain(argc, argv, nil,
            NSStringFromClass([AppDelegate class]));
    }
}
```

- Swift

在「AppDelegate.swift」的「application(\_:didFinishLaunchingWithOptions:)」方法中初始化 SDK

若需要采集应用启动数据请参考《swift-main》

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {  
    //默认为https, 如果redirect_host为http协议的话则需要调用该接口。  
    NBSAppAgent.setHttpEnabled(true)  
    NBSAppAgent.setRedirectURL("redirect_host")  
    NBSAppAgent.start(withAppID: "Your_appkey")  
    return true  
}
```

### 3、开启可视化命名（可选配置）

开启可视化命名功能，可以通过在App应用内点选的方式为「原生页面」和「操作」进行重命名并在用户体验模块中展示

- 获取 Scheme

- 在应用「设置」中的「基础设置」中选择【URL Scheme】

#### 应用设置

基础设置

安全设置

阈值设置

评分设置

请求设置

\* App名称: 演示账号

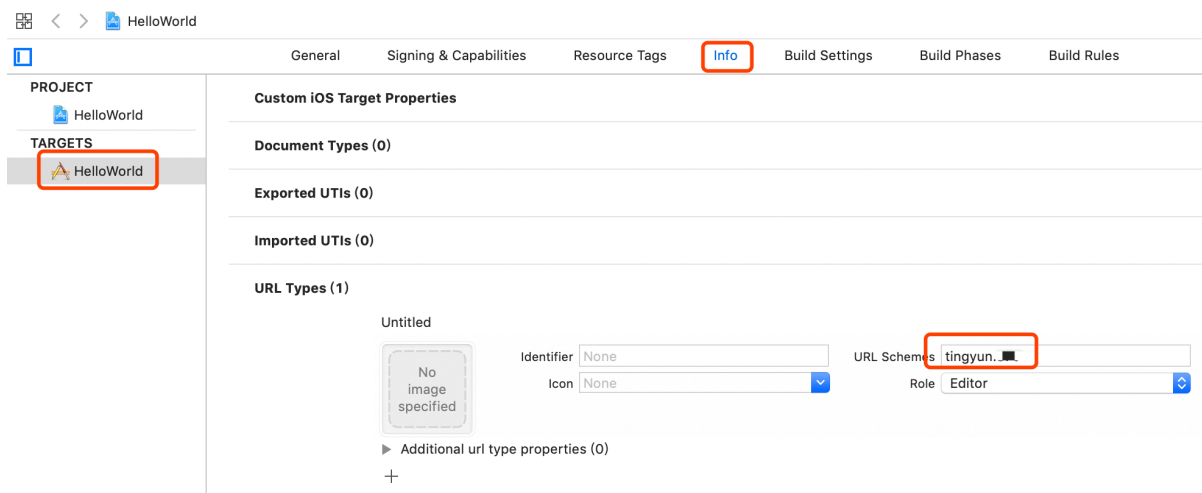
使用字母数字和中文，最长32个字符。

App Key: el40822x25w-1:8:91w025392:94761

host: wud.33unxn

URL Scheme: tingyun.803

- 将获取到的「URL Scheme」添加到TARGETS → Info → URL Scheme 中



## 4、首次启动开启模块功能开关

出于兼容性考虑，首次启动 SDK 仅在调试模式下开启全部功能模块，非调试模式下仅开启崩溃模块，可通过以下接口开启 SDK 的模块开关

- 模块功能开关接口

```
/*
    SDK首次初始化由于尚未与基调听云平台交互，默认模块开关仅开启「崩溃模块」，可以通过开启「网络模块」和「用户体验模块」来采集应用首次启动数据。
    @warning:调用该接口设置启动选项，SDK首次启动不受基调听云平台开关控制
*/
+ (void)setStartOption:(NBSOPTION)option;
```

- 代码示例

```
//在SDK初始化之前调用， 首次启动开启网络、crash采集功能、用户体验分析。
[NBSAppAgent setStartOption:NBSOption_Net|NBSOption_UI|NBSOption_Crash];
[NBSAppAgent startWithAppID:@"Your_appkey"];
```

## 5、获取用户标识

通过添加「用户标识」可在基调听云报表平台通过该标识检索到具体用户的性能问题

- 相关接口

```
/*
    设置用户标识符,不能超过64个字符，可以在任意位置多次调用（值覆盖）
    @userId:唯一标识一个用户的信息
*/
+ (void)setUserIdentifier:(NSString *)userId;
```

- 代码示例

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    //用户标识可为邮箱、手机号等能够标识用户身份的信息，如xxx@tingyun.com
    [NBSAppAgent setUserIdentifier:@"userId"];
}
```

## 6、SDK传输数据使用国密加密（可选配置）

SDK传输数据支持国密加密，调用该接口开启国密加密，默认不开启「前提：需工程中导入 NBSGKit.framework且后端需要设置开启国密（请联系技术支持设置开启后端国密加密）」

注：该接口需要在SDK初始化时调用

- 相关接口

```
/**
 * @need 传入yes,则SDK传输数据使用国密加密。
 */
+ (void)encryptionRequired:(BOOL)need;
```

- 代码示例

```
int main(int argc, char * argv[]) {
    @autoreleasepool {
        //开启国密加密
        [NBSAppAgent encryptionRequired:YES];
        //默认为https，如果redirect_host为http协议的话则需要调用该接口。
        [NBSAppAgent setHttpEnabled:YES];
        //redirect_host为平台服务器地址
        [NBSAppAgent setRedirectURL:@"redirect_host"];
        //Your_appkey从平台中获取
        [NBSAppAgent startWithAppID:@"Your_appkey"];
        return UIApplicationMain(argc, argv, nil,
        NSStringFromClass([AppDelegate class]));
    }
}
```

## 7、嵌码完整性校验

嵌码完成后，可以在控制台中查看基调听云 SDK 日志输出结果用以判断嵌码是否成功

- 日志输出

```
NBSAppAgent SDK_Version
```

```
NBSAppAgent start!
```

```
Success to connect to NBSSERVER
```