SDK部署

SDK集成

1、手动集成

- 下载并解压nbs-newlens-ios.zip
- 将tingyunApp.framework拖拽到xcode工程中（勾选☑️ **Copy itmes if needed**）
- 选择工程target -> **Build Setting** ->搜索**Other Linker Flags** , 添加编译标志**-ObjC**
- 添加一下依赖库：
  - SystemConfiguration.framework
  - CoreTelephony.framework
  - CoreGraphics.framework
  - QuartzCore.framework
  - Security.framework
  - WebKit.framework
  - libc++.tbd
  - libz.tbd

2、CocoaPods集成

在项目工程的Podfile文件中添加： **pod 'tingyunApp'**

3、Carthage集成

在项目工程的Cartfile文件中添加：**github "TingYunAPM/TingYunApp"**

初始化SDK

1、在待监测App的pch文件中引入听云App探针的头文件：

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

Swift工程在桥街头文件中引入：

```
#ifndef testswift__Bridging_Header_h
#define testswift__Bridging_Header_h
#import <tingyunApp/NBSAppAgent.h>
#endif /* testswift__Bridging_Header_h */
```

2、在嵌码项目工程main.m的main方法中添加初始化；也可以在AppDelegate.m的application:didFinishLaunchingWithOptions:（该方法中初始化会采集不到App启动时间）方法中初始化：

```
int main(int argc, char * argv[]) {
    @autoreleasepool {
        [NBSAppAgent startWithAppID:@"YOUR_APPKEY" location:YES];
        return UIApplicationMain(argc, argv, nil, NSStringFromClass([AppDelegate c
lass]));
    }
}
```

Swift工程请在AppDelegate.swift文件的application(_:didFinishLaunchingWithOptions:)方法中初始化：

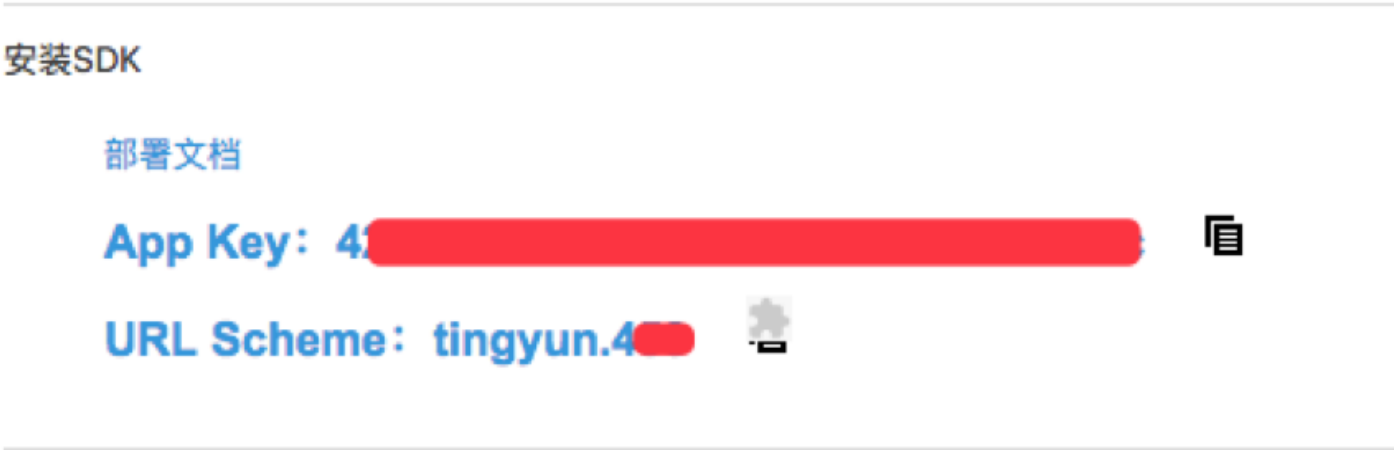
```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launc
hOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        NBSAppAgent.start(withAppID: "Your_appkey", location: true)
        return true
}
```

添加URL Scheme

将URL Scheme添加到项目中，以便我们唤醒您的应用，进行点选。

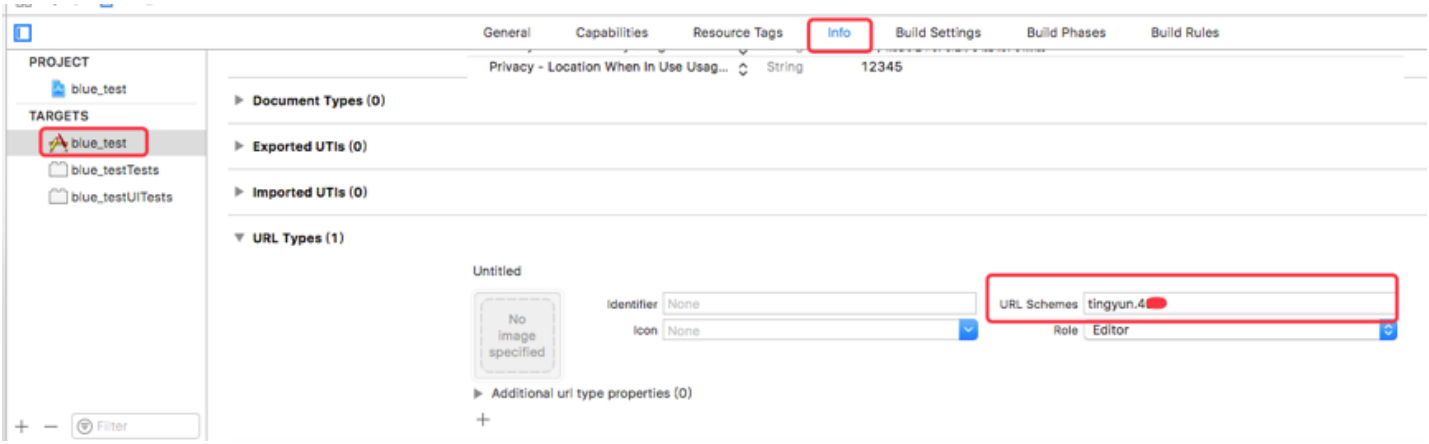
URL Scheme获取：

登录控制台，进入APP概览页 –> 点击小齿轮 –> 点击修改设置，进入到修改设置详情页面，URL Scheme就在appkey下面，格式为：tingyun.appid。



URL Scheme配置：

将获取到的URL Scheme添加到target –> info –> URL Types 。



嵌码完整性校验

嵌码完成后可以在控制台中查看听云SDK 日志输出结果：

```
NBSAppAgent  SDK_Version
---->start!
Success to connect to NBSSERVER
```

注：如果工程中嵌入了bugly，请将听云SDK放在bugly之后初始化。