# Flutter-iOS 集成文档

### 选择SDK集成方式

### 一.使用 CocoaPods 管理依赖

1. 在pubspec.yaml 文件中加入依赖

flutter\_growingio\_track:

git:

url: https://github.com/growingio/flutter-growingio-track.git

2. 引入打点所需文件

import'package:flutter\_growingio\_track/flutter\_growingio\_track.dart';

3. 添加pod 'GrowingCoreKit' 到Podfile中

4. 执行pod update,不要用--no-repo-update选项

5. 添加项目依赖库的位置在项目设置target -> 选项卡General -> Linked Frameworks and Libraries

6. 在xcode项目中加入FlutterGrowingIOTrack.h 与 FlutterGrowingIOTrack.m

（文件github地址 https://github.com/growingio/flutter-growingio-track iOS目录下）

7. 在AppDelegate中引入头文件

#import "FlutterGrowingIOTrack.h"

并在

- (BOOL)application:(UIApplication\*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions 代理方法中加入方法调用

删除删除原有的注册[GeneratedPluginRegistrant registerWithRegistry:self];

改为

   [FlutterGrowingIOTrack registerWithRegistry:self];

### 二.手动安装

1.在pubspec.yaml 文件中加入

flutter\_growingio\_track:

git:

url: <https://github.com/growingio/flutter-growingio-track.git>

引入打点所需文件

import'package:flutter\_growingio\_track/flutter\_growingio\_track.dart';

2. 获取sdk zip包

3. 解压iOS SDK压缩文件

4. 将Growing.h 和GrowingCoreKit添加到iOS工程

5. 在xcode项目中加入（文件github地址 https://github.com/growingio/flutter-growingio-track ）FlutterGrowingIOTrack.h 与 FlutterGrowingIOTrack.m 文件

6. 在AppDelegate中引入头文件

#import "FlutterGrowingIOTrack.h"

并在

- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions 代理方法中加入方法调用

删除删除原有的注册[GeneratedPluginRegistrant registerWithRegistry:self];

改为

[FlutterGrowingIOTrack registerWithRegistry:self];

# 设置URL Scheme（必选）

(1) 获取URL Scheme

• 添加新产品：登录官网 -> 点击项目选择框 -> 点击“项目管理” -> 点击“应用管理” -> 点击“新建应用”->选择添加 iOS 应用 -> 填写“应用名称“，点击下一步 ->在第二段中标黄字体。

• 现有产品：登录官网 -> 点击项目选择框 -> 点击“项目管理” -> 点击“应用管理” -> 找到对应产品的 URL Scheme

添加 URL Scheme（growing.xxxxxxxxxxxxxxxx）到项目中

添加依赖, 在项目中添加以下库文件

库名称

类型

Foundation.framework

基础依赖库

Security.framework

用于SSL连接

CoreTelephony.framework

用于读取运营商名称

SystemConfiguration.framework

用于判断网络状态

AdSupport.framework

用于来源管理激活匹配

libicucore.tbd

用于WebSocket

libsqlite3.tbd

存储日志

CoreLocation.framework

用于读取地理位置信息（如果您的app有权限）

添加完成以后, 库的引用如下: 提醒:

(optional) GrowingIO推荐您添加AdSupport.framework依赖库,用于来源管理激活匹配,有利于您更好的分析的数据

• 添加项目依赖库的位置在项目设置target -> 选项卡General -> Linked Frameworks and Libraries

1. 添加编译参数

2. 添加初始化函数

在 AppDelegate 中引入#import "Growing.h"并添加启动方法

#import "Growing.h"

- (BOOL)application:(UIApplication \*)application

didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {

...

// 启动GrowingIO

[Growing startWithAccountId:@"xxxxxxxxxxxxxxxx"]; //替换为您的ID

// 其他配置

// 开启Growing调试日志 可以开启日志

// [Growing setEnableLog:YES];

}

请确保将代码添加在上面描述的位置，添加到其他函数中或者异步 block 中可能导致数据不准确！

App Store 提交应用注意事项

如果您添加了库AdSupport.framework, GrowingIO则会启用 IDFA，所以在向 App Store 提交应用时，需要：

• 对于问题 Does this app use the Advertising Identifier (IDFA)，选择 YES。

• 对于选项Attribute this app installation to a previously served advertisement，打勾。

• 对于选项Attribute an action taken within this app to a previously served advertisement，打勾。

为什么 GrowingIO 使用 IDFA? GrowingIO 使用 IDFA 来做来源管理激活设备的精确匹配，让你更好的衡量广告效果。如果你不希望跟踪这个信息，可以选择不引入 AdSupport.framework.

至此，您的SDK安装就成功了。登录 GrowingIO 进入产品安装页面执行“数据检测”，几分钟后就可以看到数据了。