

【集成文档】大疆 Flutter SDK 集成文档

- 1. SDK 功能清单
- 2. 3. SDK 预置事件与预置属性
- 3. 集成神策分析 SDK Flutter 插件
 - 3.1. 引入插件
- 4. Flutter 插件初始化
 - 4.1. 获取项目数据接收地址
 - 4.2. 初始化 SDK
 - 4.2.1. init() 方法参数说明
- 5. 配置 Scheme
 - 5.1. 获取项目 Scheme
 - 5.2. App 中添加 Scheme
- 6. SDK 基本配置
 - 6.1. 用户关联
 - 6.2. 设置事件公共属性
 - 6.3. 记录激活事件
 - 6.4. 代码埋点追踪事件
- 7. 调试查看事件信息

神策 Flutter SDK Flutter 插件 `universal_statistics_flutter_plugin`，封装了神策 Android 和 iOS SDK 常用 API，使用此插件，可以完成代码埋点的统计和上报。

1. SDK 功能清单

一级功能	二级功能	具体功能 / 描述
标识用户	自动标识匿名用户	自动使用设备 ID 标识匿名用户
		支持替换为自定义的匿名 ID
	标识登录用户	使用登录 ID 标识登录用户
	设置用户属性	
采集数据	自动采集设备信息	自动采集默认的设备信息，屏幕宽高、系统版本号等
		支持自动采集屏幕方向
		支持自动采集经纬度信息
	全埋点	App 启动
		App 退出
	公共属性	静态公共属性
		清除公共属性
	自定义埋点	采集激活（安装）事件
		统计事件时长
		自定义代码埋点
上报数据	自动上报数据	SDK 在满足一定条件后自动发送事件数据
	手动上报数据	SDK 支持手动上报事件数据
调试功能	调试模式	开启 / 关闭调试模式
	调试日志	显示 / 关闭调试日志

Flutter SDK 目前功能清单要低于原生 SDK 提供的功能，即并未全部包含原生 SDK 的功能。若要使用原生 SDK 提供功能，可在原生端初始化 SDK。具体初始化方式可参考：[Android SDK 使用指南](#) 和 [iOS SDK 使用指南](#)。

2. 3. SDK 预置事件与预置属性

参见 [App SDK 预置事件与预置属性文档](#)

- 在使用前，请先阅读[数据模型](#)、[Android SDK 使用指南](#)和 [iOS SDK 使用指南](#)

3. 集成神策分析 SDK Flutter 插件

3.1. 引入插件

在 Flutter 项目的 *pubspec.yaml* 文件中 dependencies 里面添加 universal_statistics_flutter_plugin 依赖。

```
dependencies:  
  # 添加 flutter plugin  
  universal_statistics_flutter_plugin: ^3.0.0
```

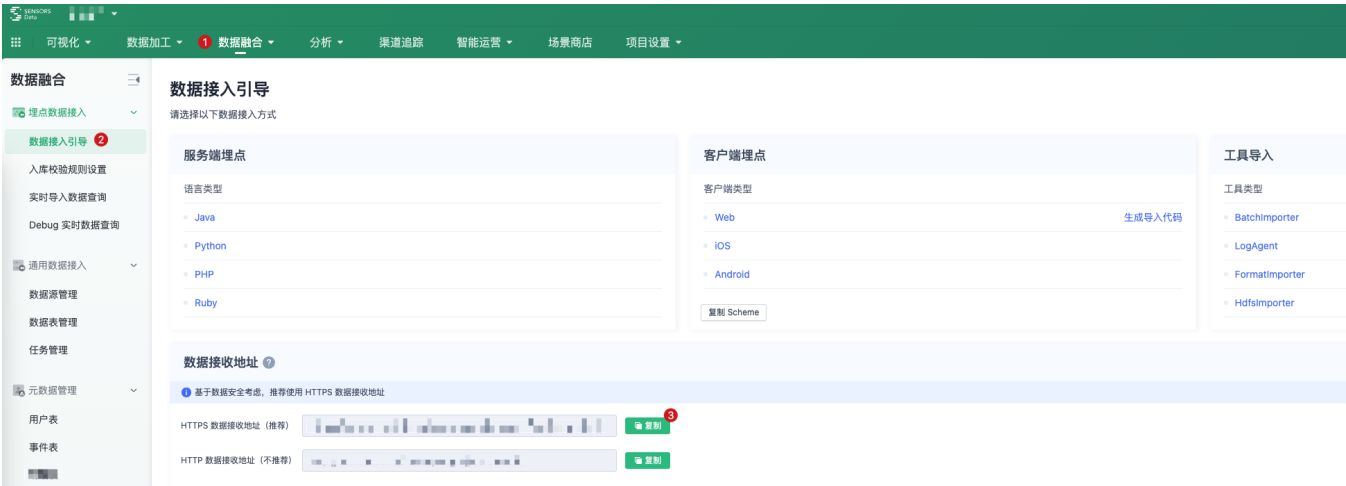
执行 *flutter pub get* 命令安装插件。

```
flutter pub get
```

4. Flutter 插件初始化

4.1. 获取项目数据接收地址

- 每个项目都有单独的数据接收地址
- 请使用[管理员](#)账户获取相应项目的数据接收地址



4.2. 初始化 SDK

自 Flutter 插件 [v2.1.0](#) 开始支持在 Flutter 端进行初始化。示例如下：

```
UniversalFlutterPlugin.init(  
  serverUrl:  
    "数据接收地址",  
  autoTrackTypes: <SAAutoTrackType>{  
    SAAutoTrackType.APP_START,  
    SAAutoTrackType.APP_END  
  },  
  networkTypes: <SANetworkType>{  
    SANetworkType.TYPE_ALL,  
  },  
  flushInterval: 10000,  
  flushBulkSize: 50,  
  enableLog: true,  
  javascriptBridge: true,  
  encrypt: true,  
  heatMap: true,  
  visualized: VisualizedConfig(autoTrack: true, properties: true),  
  android: AndroidConfig(  
    maxCacheSize: 48 * 1024 * 1024,  
    jellybean: true,  
    subProcessFlush: true),  
  ios: IOSConfig(maxCacheSize: 10000),  
  globalProperties: {'aaa': 'aaa-value', 'bbb': 'bbb-value'});
```



注意：通常原生端和 Flutter 端只要初始化一次即可。若选择在 Flutter 端初始化，就不需要再在原生端初始化了。

4.2.1. init() 方法参数说明

参数名	参数类型	参数说明	备注
serverUrl	String?	数据接收地址	
enableLog	bool	是否显示日志	
autoTrackTypes	Set<SAAutoTrackType>?	全埋点采集类型	<ul style="list-style-type: none">SAAutoTrackType.NONESAAutoTrackType.STARTSAAutoTrackType.END
flushInterval	int	设置两次事件上报的最小间隔	默认 15 秒，最小 5 秒，单位毫秒
flushBulkSize	int	设置本地缓存触发 flush 的最大条目数	默认 100，最小 50
networkTypes	Set<SANetworkType>?	设置上报网络情况	TYPE_NONE, TYPE_2G, TYPE_3G, TYPE_4G, TYPE_WIFI, TYPE_5G, TYPE_ALL
encrypt	bool	是否开启加密	仅支持 RSA + AES
javascriptBridge	bool	是否支持 H5 打通	默认 false
android	AndroidConfig?	Android 端特有配置 <ul style="list-style-type: none">subProcessFlush: 是否支持子进程上报数据jellybean: 打通是否支持 API level 16 及以下的版本maxCacheSize: 最大缓存数，单位 byte 默认为:32M，即 32* 1024 * 1024	android:{ subProcessFlush:true, jellybean:true, maxCacheSize:48 * 1024*1024 }
ios	IOSConfig?	iOS 端特有配置 <ul style="list-style-type: none">maxCacheSize: 最大缓存数，单位条。默认值：10000	ios:{ maxCacheSize:10000 }
globalProperties	Map?	配置全局公共属性	其优先级低于静态公共属性



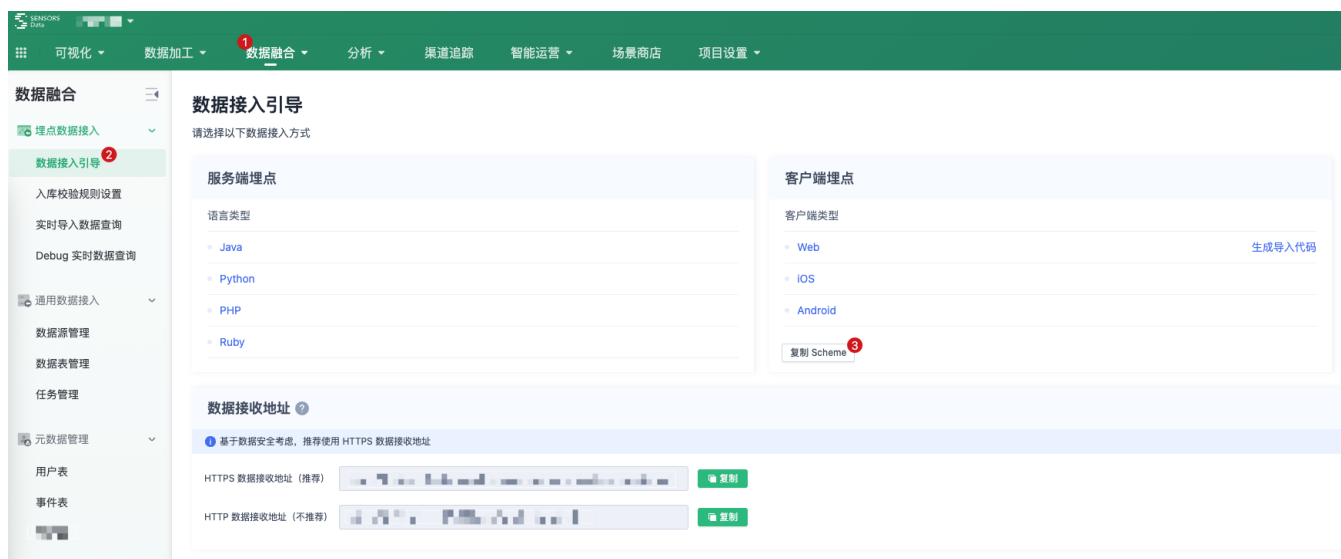
1. 若 App 有合规需求，可参考 [合规说明](#)；
2. 以上是在 Flutter 代码中进行初始化。老版本的插件不支持这种初始化方式，若需要在原生端初始化，请参考 [Android SDK 使用指南](#)和 [iOS SDK 使用指南](#)。

5. 配置 Scheme

5.1. 获取项目 Scheme



- 项目的 Scheme 需要管理员账户进行获取
- App 工程中可以同时配置多个项目的 Scheme



5.2. App 中添加 Scheme

在使用神策系统中的 Debug 实时查看、可视化全埋点等需要扫码的功能时，用于拉起页面。针对 Android 和 iOS 平台，配置方式不一样：

- Android 平台配置 Scheme ([此处需要配置链接](#))
- iOS 平台配置 Scheme ([此处需要配置链接](#))

6. SDK 基本配置

6.1. 用户关联

用户关联是为了对用户进行唯一标识，提高用户行为分析的准确性。目前神策提供了简易用户关联和全域用户关联分为用于支撑不同的业务场景。

- [简易用户关联 \(IDM 2.0\)](#)
- [全域用户关联 \(IDM 3.0\)](#)

6.2. 设置事件公共属性

对于所有事件都需要添加的属性，初始化 SDK 后，可以通过 `registerSuperProperties()` 将属性注册为公共属性。详细使用文档参见[基础 API 功能介绍](#)。

6.3. 记录激活事件

可以调用 `trackAppInstall()` 方法记录激活事件，多次调用此方法只会在第一次调用时触发激活事件。详细使用文档参见[渠道追踪与广告](#)。

6.4. 代码埋点追踪事件

SDK 初始化后，可以通过 `track()` 方法追踪用户行为事件，并为事件添加自定义属性。详细使用文档参见[基础 API 功能介绍](#)。

7. 调试查看事件信息

可以在 `init()` 方法初始化时 `enableLog` 设置参数为 `true`。或者调用

```
import 'package:universal_statistics_flutter_plugin/universal_statistics_flutter_plugin.dart';

UniversalFlutterPlugin.enableLog(true/false) //方法来开启或关闭日志。
```

在 **Logcat (Android)** 或 **Xcode (iOS)** 中筛选 **SA** 关键词：

- 埋点事件触发成功时，SDK 会输出 **track event** 开头的事件数据
- 埋点事件触发失败时，SDK 会输出相应的错误原因
- 事件数据上报成功时，SDK 会输出 **valid message** 字段开头的事件数据
- 事件数据上报失败时，SDK 会输出 **invalid message** 字段开头的事件数据并输出错误原因