

千寻差分数据 iOS SDK

开发指南

V1.3.0

千寻位置网络有限公司

2016 年 12 月·上海

法律声明

版权所有© 2016，千寻位置网络有限公司。保留一切法律权利。本文档包含的所有内容除特别声明之外，版权均属于千寻位置网络有限公司所有，受《中华人民共和国著作权法》及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式(电子或机械,包括影印)或理由对该文档或其包含的任何产品、服务、信息、材料的任何部分进行使用、复制、修改、抄录、传播或与其它产品捆绑使用、销售，否则将视为侵权，本公司必依法追究其法律责任。本文档并不代表供应商或其代理的承诺，千寻位置网络有限公司可在不作任何申明的情况下对本文档内容进行修改。本文档中提到的其它公司及其产品的商标所有权属于该商标的所有者。

千寻位置网络有限公司

联系邮箱：service@qxwz.com

官方网站：www.qxwz.com

目 录

第一章 概述.....	4
1.1 产品简介	4
第二章 开发指南.....	5
2.1 使用流程	5
2.2 申请 AppKey、AppSecret	6
2.3 配置工程	6
2.3.1 注意事项	错误! 未定义书签。
2.3.2 配置步骤	6
第三章 SDK 接口说明	8
3.1 示例代码	8
3.2 API 说明	9
3.2.1 QXWZSDK 类	9
3.2.2 QXWZRtcmManager 类	10
3.2.3 QXWZAccount 类	10
3.2.4 QXWZRtcmManagerDelegate 代理	11
3.2.5 QXWZLocation 坐标类	11
3.2.6 QXWZRtcmEnum 枚举类	11

第一章 概述

1.1 产品简介

千寻位置网差分数据 iOS SDK 为基于 iOS 平台的移动端应用提供自动注册差分账号和自动获取差分数据的服务接口,专注于为广大开发者提供差分数据服务。

该 SDK 免费对外开放,适用于大批量高精度终端的差分账号智能分配和差分数据获取的繁琐操作。使用该 SDK 获取差分数据后,差分数据传输、芯片写入工作完全交由开发者完成,为开发者提供最大灵活性。用户在获取这些服务的同时,不用考虑数据存储,数据安全和繁琐的网络交互等细节。

千寻位置网差分数据 SDK 主要提供以下功能:

用户使用自己的高精度芯片差分算法,使用 SDK 可以获取差分数据,用户可以使用这些差分数据灌进相关定位芯片或算法。

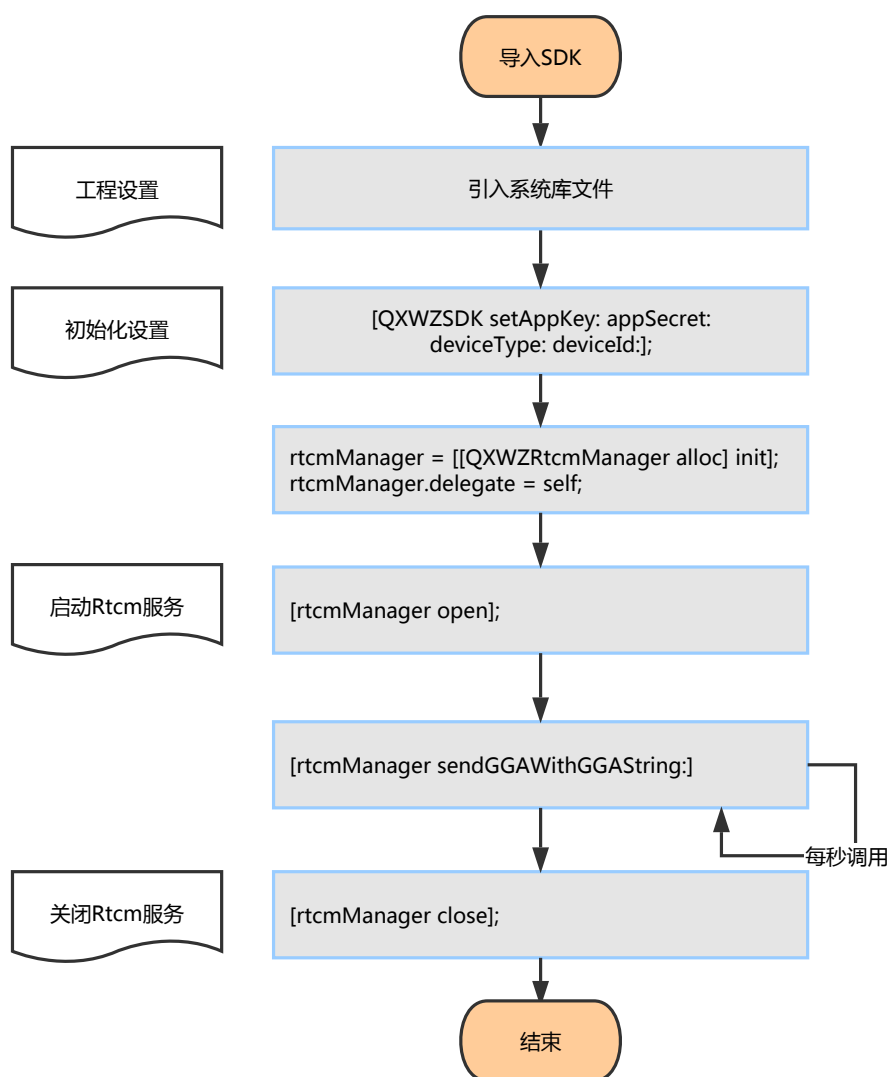
使用 SDK,需要申请 AppKey 和 AppSecret,并确保有足够的差分账号配额。请在千寻位置网官网(qxwz.com)注册账号,按照[帮助文档](#)完成注册、认证、应用创建、AppKey 和 AppSecret 获取以及差分账号管理工作。

第二章 开发指南

2.1 使用流程

SDK 使用流程大体如下：

需要先申请相关的 AppKey，获取相关的密钥。然后工程引入 SDK 包，调用对应的 API 接口完成操作。



2.2 申请 AppKey、AppSecret

AppKey、AppSecret 由千寻位置网提供，获取办法请见[在线帮助文档](https://www.qxwz.com/help-document-settled-platform.html#link-0)

(<https://www.qxwz.com/help-document-settled-platform.html#link-0>)。

2.3 配置工程

2.3.1 配置步骤

1. 下载开发包并解压。解压后得到两个文件，分别是

- QXWZRtcmSDK.framework
- QXWZSDK_CoreData.bundle

2. 引入差分静态包

在 Targets->Build Phases->Link Binary With Libraries 配置项点击 “+”，弹出添加列表后，点击 “Add Other...” ，添加 QXWZRtcmSDK.framework 到工程中。

3. 引入系统库文件

在 Targets->General->Linked Frameworks and Libraries 中点击 “+”，弹出添加列表后添加如下系统库：

- CoreLocation.framework
- CoreMotion.framework
- CoreData.framework
- SystemConfiguration.framework
- libc++.tdb
- libz.1.2.5.tdb

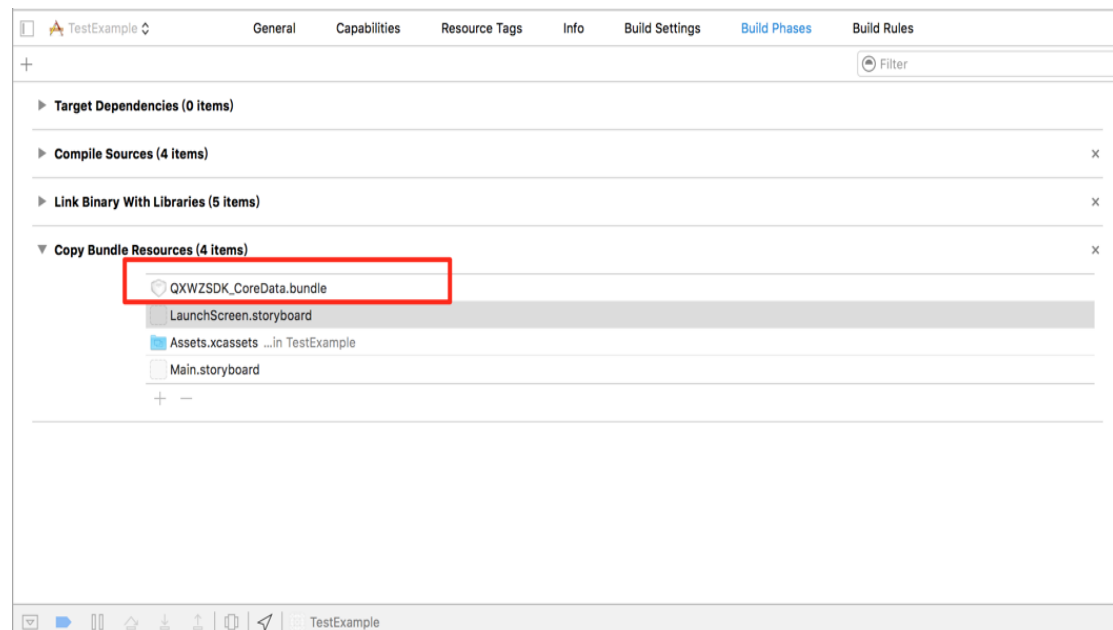
上述(2)(3)两步完成之后，Link Binary With Libraries 应该如截图：

▼ Linked Frameworks and Libraries	
Name	Status
libz.1.2.5.tbd	Required
libc++.tbd	Required
CoreMotion.framework	Required
CoreLocation.framework	Required
SystemConfiguration.framework	Required
CoreData.framework	Required
QXWZRtcmSDK.framework	Required

4. 引入资源文件

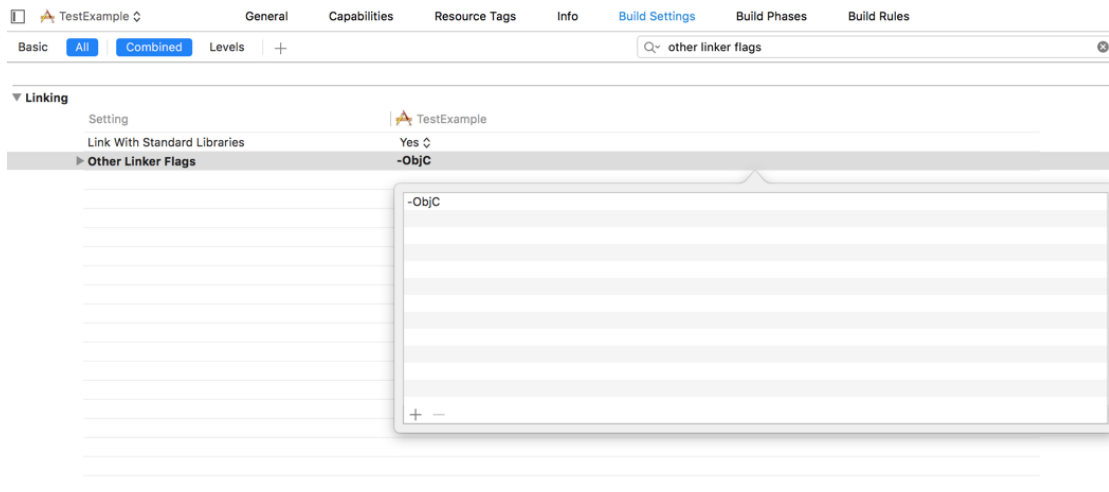
在 TARGETS->Build Phases->Copy Bundle Resources 中点击 “+”，弹出添加列表

后添加如下资源文件：QXWZSDK_CoreData.bundle



5. 添加 Other Linker Flags

左侧目录中选中工程名，在 Targets->Building Settings->Linking->Other Linker Flags 中加上-ObjC



第三章 SDK 接口说明

3.1 示例代码

```
#import <QXWZRtcmSDK/QXWZRtcmManager.h>
#import <QXWZRtcmSDK/QXWZRtcmEnum.h>
#import <QXWZRtcmSDK/QXWZAccount.h>
#import <QXWZRtcmSDK/QXWZSDK.h>

@interface TestViewController()<QXWZRtcmManagerDelegate>
@property (nonatomic, strong) QXWZRtcmManager *rtcmManager;
@end

@implementation TestViewController

- (instancetype)init
{
    self = [super init];
    if (self) {
        [QXWZSDK setAppKey:@"从千寻位置官网申请的 appKey" appSecret:@"从千寻位置官网申请的 appSecret" deviceType: @"设备类型" deviceId:@"设备唯一标识"];
        _rtcmManager = [[QXWZRtcmManager alloc] init];
    }
}
```

```

        _rtcmManager.delegate = self;
    }
    return self;
}

// 启动 Rtc SDK
- (void)start{
    [_rtcmManager open];
}

// 关闭 Rtc SDK
- (void)stop{
    [_rtcmManager close];
}

#pragma mark - QXWZRtcManagerDelegate

- (void)qxwz_rtcManager:(QXWZRtcManager *)manager receivedRtcData:(NSData *)data{
    // TODO: 获取到差分数据时回调
}

- (void)qxwz_rtcManager:(QXWZRtcManager *)manager statusUpdate:(QXWZRtcStatus)statusCode{
    // TODO: 差分服务状态发送变化时回调
}

@end

```

3.2 API 说明

3.2.1 QXWZSDK 类

用于设置 appKey、appSecret、deviceType、deviceId，首先要调用 setAppKey:appSecret:

deviceType:deviceId 方法。

方法/变量	描述
setAppKey:appSecret:deviceType:deviceId	设置从千寻位置官网申请的 appKey、appSecret，设置设备的类型和设备唯一标识，该方法要在所有 API 之前调用。
sdkVersion	查看当前 SDK 的版本号

appKey	查看设置的 appKey
deviceType	查看设置的设备类型
deviceId	查看设备的设备唯一标识

3.2.2 QXWZRtcmManager 类

iOS Rtcm SDK 主入口，连接 Ntrip 服务器，发送和获取数据。

方法/变量	描述
delegate	设置需要处理回调的类的对象。
getQXWZAccount	获取差分账户信息。
open	启动差分服务。
close	关闭差分服务。
sendGGAWithGGAString	发送 GGA 字符串来获取差分数据，超过两分钟不调用该方法，SDK 将不会产生差分数据。
sendGGAWithLatitude:longitude	发送经纬度来获取差分数据，超过两分钟不调用该方法，SDK 将不会产生差分数据。 与 sendGGAWithGGAString 只需要调用一个即可。

3.2.3 QXWZAccount 类

方法/变量	描述
appKey	千寻位置官网申请的 appKey
deviceId	查看设置的设备唯一标识
deviceType	查看设置的设备类型
serviceType	查看 SDK 服务类型
expireTime	查看账户到期时间
ntripUserName	查看差分账户用户名
ntripPassword	查看差分账户密码

3.2.4 QXWZRtcmManagerDelegate 代理

该代理里面包含两个回调方法。

方法/变量	描述
- (void)qxwz_rtcManager:(QXWZRtcmManager *)manager receivedRtcmData:(NSData *)data;	当获取到差分数据时回调该方法。
- (void)qxwz_rtcManager:(QXWZRtcmManager *)manager statusUpdate:(QXWZRtcmStatus)statusCode;	当差分状态发生变化时回调。 statusCode 为枚举类型，具体的取值范围请参照 QXWZRtcmEnum； message 为返回的状态描述。

3.2.5 QXWZLocation 坐标类

差分 SDK 可不关注该类。

方法/变量	描述
latitude	纬度
longitude	经度
timestamp	惯导抛出经纬度的时间

3.2.6 QXWZRtcmEnum 枚举类

差分运行状态枚举：

QXWZRtcmStatus 枚举 Code	message 描述
QXWZRtcmStatus_NTRIP_CONNECTED	ntrip 连接到服务器
QXWZRtcmStatus_NTRIP_DISCONNECTED	ntrip 连接断开
QXWZRtcmStatus_APPKEY_IDENTIFY_FAILURE	appKey 验证无效
QXWZRtcmStatus_APPKEY_IDENTIFY_SUCCESS	appKey 验证成功
QXWZRtcmStatus_NETWORK_UNAVAILABEL	网络不可用
QXWZRtcmStatus_NTRIP_USER_FAIL	appKey,appSecret 验证失败
QXWZRtcmStatus_NTRIP_USER_MAX	appKey 用户已经达到上限
QXWZRtcmStatus_NTRIP_CONNECTING	正在连接 ntrip 服务器
QXWZRtcmStatus_NTRIP_RECEIVING_DATA	ntrip 正在接受数据
QXWZRtcmStatus_OPENAPI_PARAM_MISSING	缺少参数
QXWZRtcmStatus_OPENAPI_ACCOUNT_NOT_EXIST	账号不存在
QXWZRtcmStatus_OPENAPI_DUPLICATE_ACCOUNT	账号重复

QXWZRtcmStatus_OPENAPI_INCORRECT_PASSWORD	错误密码
QXWZRtcmStatus_OPENAPI_DISABLED_ACCOUNT	账号未激活
QXWZRtcmStatus_OPENAPI_NO_AVAILABLE_ACCOUNT	没有有效的账号
QXWZRtcmStatus_OPENAPI_NO_RELATED_POPUSER	POPUser 不存在
QXWZRtcmStatus_OPENAPI_SYSTEM_ERROR	服务端内部错误
QXWZRtcmStatus_OPENAPI_ACCOUNT_EXPIRED	账号已过期需续费
QXWZRtcmStatus_OPENAPI_ACCOUNT_WILL_EXPIRE	账号即将到期