# SeekcyBeacon SDK for iOS 集成指南

Seekcy官网上，有SeekcyBeacon相关的所有指南、API、教程等全部文档。本文档的更新版本，都会及时发布到该网站上。[下载链接](https://github.com/Seekcy/SeekcyBeacon_SDK_IOS)

## 产品功能说明

目前我们的iOS SDK 主要提供以下功能

1. 扫描周围的SeekcyBeacon
2. 连接、配置SeekcyBeacon
3. 区域监听SeekcyBeacon：进入、离开

## SDK集成

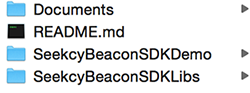
### SeekcyBeacon\_SKD\_IOS-master.zip集成压缩包内容

SeekcyBeaconSDKLibs：.framework库

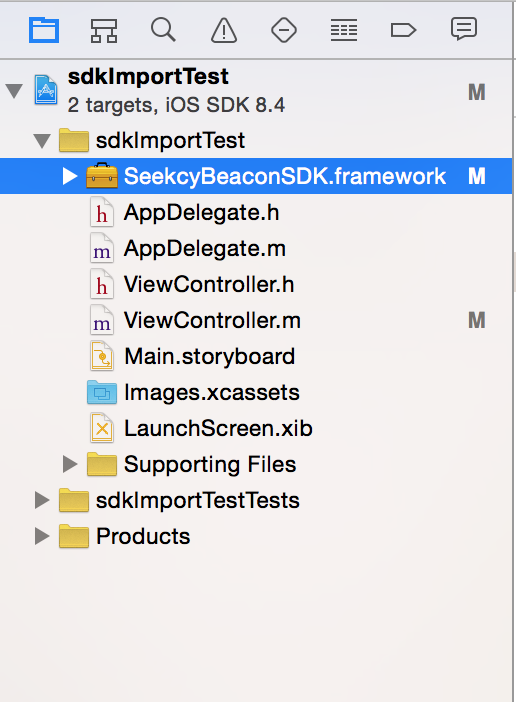
SeekcyBeaconSDKDemo：一个完整的IOS项目，演示SeekcyBeacon SDK的基本用法，可以用作参考

Documents：SDK API、SDK使用说明

README.md：版本更新信息



### 导入SeekcyBeaconSDK.framework



引入: CoreMotion.framework

libsqlite3.dylib

libc++.dylib

libz.dylib

libxml2.dylib

CoreLocation.framework

Accelerate.framework

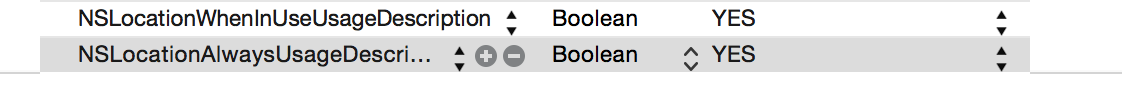
CoreGraphics.framework

QuartzCore.framework

CoreBluetooth.framework

### 配置plist，解决iOS8定位权限不足的问题：

在plist中添加2个key：NSLocationWhenInUseUsageDescription、NSLocationAlwaysUsageDescription



### Build Seeting 中 Other Linker Flags 添加 –ObjC

### Xcode 7 在Build Seeting 中设置 Enable Bitcode = NO

## 示例代码

@import SeekcyBeaconSDK;

### 扫描

（1）SeekcyBeacon防蹭用

// 设置防蹭用解密密钥

[SKYBeaconManager sharedDefaults].seekcyDecryptKey = @"12B52346AD4529166665579539C04181B2E3F58C232641D744223EED5932409D";

注意：

请在SDK启动前设置；防蹭用密钥必须为32bytes长度的十六进制String类型，密钥由Seekcy提供，并且配合支持防蹭用的SeekcyBeacon使用

（2）扫描

@interface ViewController ()<SKYBeaconManagerScanDelegate>

- (void)viewDidAppear:(BOOL)animated{

[super viewDidAppear:YES];

// 扫描

NSArray \*scanUUIDArray = @[[[SKYBeaconScan alloc] initWithuuid:@"E2C56DB5-DFFB-48D2-B060-D0F5A71096E0" name:@"apple"],[[SKYBeaconScan alloc] initWithuuid:@"FDA50693-A4E2-4FB1-AFCF-C6EB07647825" name:@"寻息摇一摇"],[[SKYBeaconScan alloc] initWithuuid:@"C91ABDBE-DF54-4501-A3AA-D7BDF1FD2E1D" name:@"aaaaa"]];

[SKYBeaconManager sharedDefaults].scanBeaconTimeInterval = 1.2;

[[SKYBeaconManager sharedDefaults] startScanForSKYBeaconWithDelegate:self uuids:scanUUIDArray distinctionMutipleID:NO isReturnValueEncapsulation:NO];

}

delegate返回结果

全部Beacon，（单id，多id beaon一起返回）:

// 返回全部 beacon

- (void)skyBeaconManagerCompletionScanWithBeacons:(NSArray \*)beascons error:(SKYBeaconManagerError \*)error{

// do sth...

}

注：单id、多id一起返回的时候，是以键值对组成的数组，而不是SKYBeacon或SKYBeaconMutipleID对象。可以根据键值对自己封装成SKYBeacon或SKYBeaconMutipleID对象。

单id， 多id 分开返回:

// 返回单id beacon

- (void)skyBeaconManagerCompletionScanWithSingleIDBeacons:(NSArray \*)beascons error:(SKYBeaconManagerError \*)error{

if(!error){

[self.beaconScanedArraySingleID removeAllObjects];

[self.beaconScanedArraySingleID addObjectsFromArray:beascons];

[self.myTableView reloadData];

}

else{

// do sth...

}

}

// 返回多id beacon

- (void)skyBeaconManagerCompletionScanWithMutipleIDBeacons:(NSArray \*)beascons error:(SKYBeaconManagerError \*)error{

if(!error){

[self.beaconScanedArrayMutipleID removeAllObjects];

[self.beaconScanedArrayMutipleID addObjectsFromArray:beascons];

[self.myTableView reloadData];

}

else{

// do sth...

}

}

### 连接

@interface MutipleIDDetailViewController ()<SKYBeaconManagerConfigurationDelegate>

// 单id

// 这里的 self.detailBeacon 是SKYBeacon 对象

[[SKYBeaconManager sharedDefaults] connectSingleIDBeacon:self.detailBeacon delegate:self];

// 多id

// 这里的 self.detailBeacon 是SKYBeaconMutipleID对象

[[SKYBeaconManager sharedDefaults] connectMutipleIDBeacon:self.detailBeacon delegate:self];

#pragma mark – SKYBeaconManagerConfigurationDelegate

// 多id delegate返回

- (void)skyBeaconManagerConnectResultMutipleIDBeacon:(SKYBeaconMutipleID \*)beacon error:(NSError \*)error{

if(error == nil){

}

else{

NSLog(@"连接失败");

}

}

// 单id delegate返回

- (void)skyBeaconManagerConnectResultSingleIDBeacon:(SKYBeacon \*)beacon error:(NSError \*)error{

if(error == nil){

}

else{

NSLog(@"连接失败");

}

}

### 取消连接

[[SKYBeaconManager sharedDefaults] cancelBeaconConnection:self.detailBeacon completion:^(BOOL complete, NSError \*error) {

if(complete){

// 取消连接成功

}

else{

// 取消连接失败

}

}];

### 配置(单id)

// 这里的 beaconToConfig 是 SKYBeaconConfigSingleID对象

[[SKYBeaconManager sharedDefaults] writeSingleIDBeaconValues:beaconToConfig completion:^(NSError \*error) {

// 配置时的错误

if(error != nil){

}

// 配置成功

else{

}

}];

### 配置(多id)

// 这里的 beaconToConfig 是 SKYBeaconConfigMutipleID 对象

[[SKYBeaconManager sharedDefaults] writeMutipleIDBeaconValues:beaconToConfig completion:^(NSError \*error) {

// 配置时的错误

if(error != nil){

}

// 配置成功

else{

}

}];

### 区域监听

@interface MonitorViewController ()<SKYBeaconManagerMonitorDelegate>

self.monitorBeacons = @[[[SKYBeaconRegion alloc] initWithProximityUUID:[[NSUUID alloc] initWithUUIDString:@"E2C56DB5-DFFB-48D2-B060-D0F5A71096E0"] identifier:@"苹果默认"],

[[SKYBeaconRegion alloc] initWithProximityUUID:[[NSUUID alloc] initWithUUIDString:@"FDA50693-A4E2-4FB1-AFCF-C6EB07647825"] identifier:@"苹果默认"]];

// 监听

[[SKYBeaconManager sharedDefaults] startMonitoringForSKYBeaconRegions:self.monitorBeacons delegate:self];

delegate返回进出入区域：

// 进

- (void)skyBeaconManagerDidEnterRegion:(CLRegion \*)region{

}

// 出

- (void)skyBeaconManagerDidExitRegion:(CLRegion \*)region{

}

### Ranging

delegate返回结果

@interface RangingViewController ()<SKYBeaconManagerRangingDelegate >

self.rangingBeacons = @[[[SKYBeaconRegion alloc] initWithProximityUUID:[[NSUUID alloc] initWithUUIDString:@"E2C56DB5-DFFB-48D2-B060-D0F5A71096E0"] identifier:@"苹果默认"],

[[SKYBeaconRegion alloc] initWithProximityUUID:[[NSUUID alloc] initWithUUIDString:@"FDA50693-A4E2-4FB1-AFCF-C6EB07647825"] identifier:@"苹果默认1"]];

[[SKYBeaconManager sharedDefaults] startRangingSKYBeaconsInRegions:self.rangingBeacons

delegate:self];

#pragma mark - SKYBeaconManagerRangingDelegate

- (void)skyBeaconManagerDidRangeBeacons:(NSArray \*)beacons inRegion:(SKYBeaconRegion \*)region{

}

### 推送

@import SeekcyBeaconSDK;

@interface PushListViewController ()<SKYBeaconManagerRangingDelegate>

NSMutableArray \*rangingArray = [[NSMutableArray alloc] init];

SKYBeaconRegion \*region = [[SKYBeaconRegion alloc] initWithProximityUUID:[[NSUUID alloc] initWithUUIDString:@"FDA50693-A4E2-4FB1-AFCF-C6EB07647825"] major:1 identifier:@"wx\_1"];

[rangingArray addObject:region];

// 不设置的话，默认 -59

[[SKYBeaconManager sharedDefaults] setRangedNearByPhoneMeasuredPower:-60];

// 不设置的话，SKYBeaconNearbyThresholdNEAR\_1\_METER

[[SKYBeaconManager sharedDefaults] setRangedNearByTriggerThreshold:SKYBeaconNearbyThresholdNEAR\_3\_METER];

[[SKYBeaconManager sharedDefaults] startRangedNearBySKYBeaconsInRegions:rangingArray completion:^(BOOL complete, NSError \*error, NSArray \*beacons) {

// beacons 是 SKYBeacon 组成的数据

}];

## 总结

至此，您已完成SDK整合进您项目的全部工作。更多技术细节，请参考[完整的SDK API文档](http://seekcy.github.io/SeekcyBeacon_SDK_IOS)，以及我们的[示例工程](https://github.com/Seekcy/SeekcyBeacon_SDK_IOS/tree/master/SeekcyBeaconSDKDemo)