

# 高德地图 iOS 室内定位 SDK 开发指南

V6.9

高德软件有限公司  
17 年 4 月 12 日·北京

## 法律声明

版权所有©2014，高德集团。

保留一切权利。

本文档包含的所有内容除特别声明之外，版权均属于高德集团所有，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）翻印或转载本文档的任何部分，否则将视为侵权，高德集团保留依法追究其法律责任的权利。

高德地图 API 的一切有关权利属于高德集团所有。

本文档并不代表供应商或其代理的承诺，高德集团可在不作任何声明的情况下对本文档内容进行修改。

本文档中所涉及的软件产品及其后续升级产品均由高德集团制作并负责全权销售。

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。

高德地图

联系邮箱：[api@autonavi.com](mailto:api@autonavi.com)

技术交流论坛：[bbs.amap.com](http://bbs.amap.com)

官方微博：<http://weibo.com/gaodedituapi>

高德地图 API 欢迎用户的任何建议或意见。

# 目 录

1. 概述 .....	2
1. 室内定位 SDK .....	3
2. 面向的读者 .....	3
3. 申请 API Key .....	3
4. 兼容性 .....	3
5. 给开发者的建议 .....	3
2. 入门指南 .....	4
2.1. 获取最新版本的 Xcode .....	4
2.2. 从 LBS 下载室内定位 SDK .....	4
2.3. 获取高德 Key .....	4
2.4. Hello , AMapIndoorLocation! .....	4
2.5. 连接 iOS 设备 .....	5
2.6. 构建和运行您的应用 .....	5
2.7. 后续步骤 .....	5
3. 创建工程 .....	6
3.1. 获取 key .....	6
3.1.1. 如何申请 Key .....	6
3.1.2. 如何获取 Bundle Identifier .....	7
3.2. 环境配置 .....	8
4. 获取室内位置 .....	10
4.2. 构造定位对象 .....	10
4.2. 设置高德地图 API KEY .....	11
4.3. 指定定位建筑 .....	12
4.5. 启动定位 .....	13
4.6. 停止定位 .....	13
4.7. 获取版本号 .....	14
4.8. 定位成功事件 .....	14
4.9. 定位失败事件 .....	15
4.10. 定位结果 .....	16
5. 实用工具 .....	17
5.1. 定位错误码 .....	17
6. 示例中心 .....	20
6.1. 高德室内定位 SDK 示例 Demo 工程 .....	20
6.2. 开发指南中的代码段 .....	21
7. 示例代码 .....	21

# 1. 概述

## 1. 室内定位 SDK

iOS 室内定位 SDK 是一套简单的 LBS 室内定位接口，通过定位 SDK 开发者可以迅速为应用程序实现室内定位功能。您可以单独使用室内定位 SDK，也可以结合高德 iOS 室内地图 SDK。

## 2. 面向的读者

高德地图 iOS 室内定位 SDK 是提供给具有一定 iOS 编程经验和了解面向对象概念的读者使用的。此外，读者还应该对地图产品有一定的了解。用户在使用中遇到任何问题，都可以通过官网提供的 QQ 群或问答社区反馈给我们。

## 3. 申请 API Key

为保证服务可以正常使用，您需要注册成为开发者并申请 Key。每个帐户，最多可以申请 10 个 Key。申请流程请参照[获取密钥](#)。

## 4. 兼容性

支持 iOS 8.0 及以上系统。

## 5. 给开发者的建议

- iOS 室内定位只有蓝牙定位，没有 WIFI 定位。
- 只有在线定位。
- iOS 室内定位默认都是开启 PDR。
- 气压计会根据实际情况检测。
- iOS 室内定位需要用户开启定位权限。
- 服务器地址默认是 LBS 正式服务器地址。
- iOS 室内定位从版本 6.7 之后不再需要设置蓝牙 UUID。

## 2. 入门指南

### 2.1. 获取最新版本的 Xcode

使用 高德 iOS SDK 开发项目，您需要 6.3 版或更高版本的 [Xcode](#)。

### 2.2. 从 LBS 下载室内定位 SDK

从 <http://lbs.amap.com/api/ios-indoorlocation-sdk/download/> 下载室内定位 SDK

### 2.3. 获取高德 Key

请前往[高德开放平台控制台](#)申请 iOS Key。

### 2.4. Hello , AMapIndoorLocation!

#### 1.配置 Info.plist 文件

由于需要申请定位权限，在 Info.plist 中加入 `NSLocationAlwaysUsageDescription` 字段。

#### 2.配置高德 Key

将上面步骤获取的高德 Key，配置到代码中，以保证定位功能的正常运行。代码如下：

```
#import <OnlineLocationSDK/OnlineLocationSDK.h> //引用室内定位模块头文件
.....

@property(nonatomic, strong) OnlineLocation* indoor; //室内定位实例
.....

实例    self.indoor = [[OnlineLocation alloc] initWithDelegate:self]; //获取定位

        self.indoor.key = @"您的 KEY";
        self.indoor.buildingId = @"室内建筑物 POIID"; //设置当前的建筑物
        [self.indoor startLocation]; //开始定位
```

## 2.5. 连接 iOS 设备

了解您的应用实际运行情况的最简单方法是将 iOS 设备与计算机相连。在手机上信任您的开发者证书，并开启定位。

## 2.6. 构建和运行您的应用

在 Xcode 中，点击 Product 菜单中 Run 选项（或 Run 按钮图标）运行您的应用。

## 2.7. 后续步骤

您可能想观摩一些示例代码，可以前往[示例中心](#)下载官方示例 Demo。

或者您也可以前往[开发指南](#)阅读更多相关内容。

## 3. 创建工程

### 3.1. 获取 key

#### 3.1.1. 如何申请 Key

1、进入[控制台](#)，创建一个新应用。如果您之前已经创建过应用，可直接跳过这个步骤。



2、在创建的应用上点击"添加新 Key"按钮，在弹出的对话框中，依次：输入应用名称，选择绑定的服务为“iOS 平台 SDK”，输入安全码 Bundle ID（获取方法请参考：[如何获取 Bundle Identifier](#)），如下图所示：

⊕ 为test添加Key

×

\* Key名称:

navi\_ios

命名规范

\* 服务平台:

☐ Android平台SDK

☒ iOS平台SDK

☐ WinPhone SDK

☐ JavaScript API

☐ Web服务API

☐ 智能硬件定位

可使用服务:

iOS地图SDK

iOS定位SDK

iOS导航SDK

iOS室内地图SDK

iOS室内定位SDK

\* 安全码Bundle ID:

com.autonavi.officialDemoNavi

☒ 我已阅读 高德地图API服务条款

取消

提交

在阅读完高德地图 API 服务条款后，勾选此选项，点击“提交”，完成 Key 的申请，此时您可以在所创建的应用下面看到刚申请的 Key 了。

### 3.1.2. 如何获取 Bundle Identifier

以下为您提供两种获取 Bundle Identifier 的方法。

方法一

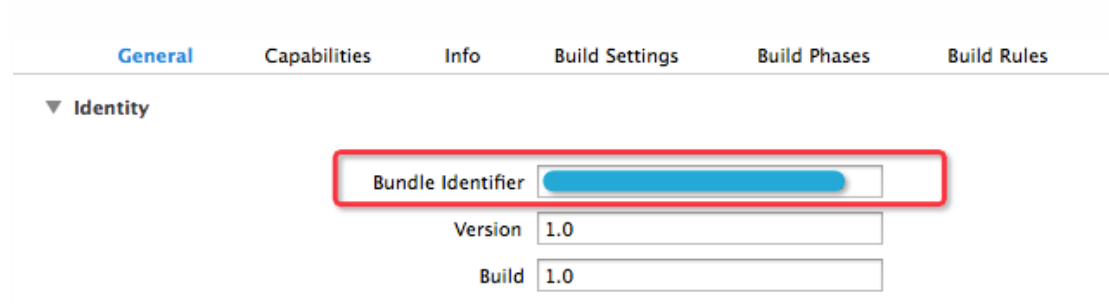
通过代码获取，代码如下所示：

```
1 NSString *bundleIdentifier = [[NSBundle mainBundle] bundleIdentifier];
```

方法二



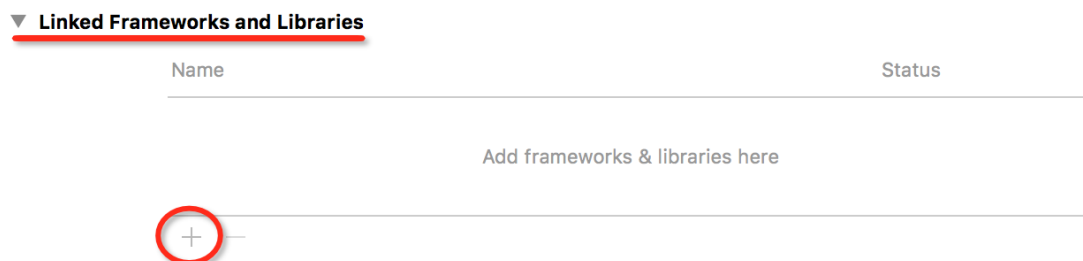
Xcode 切换到 General 标签，查看 Bundle Identifier，如下图所示：



## 3.2. 环境配置

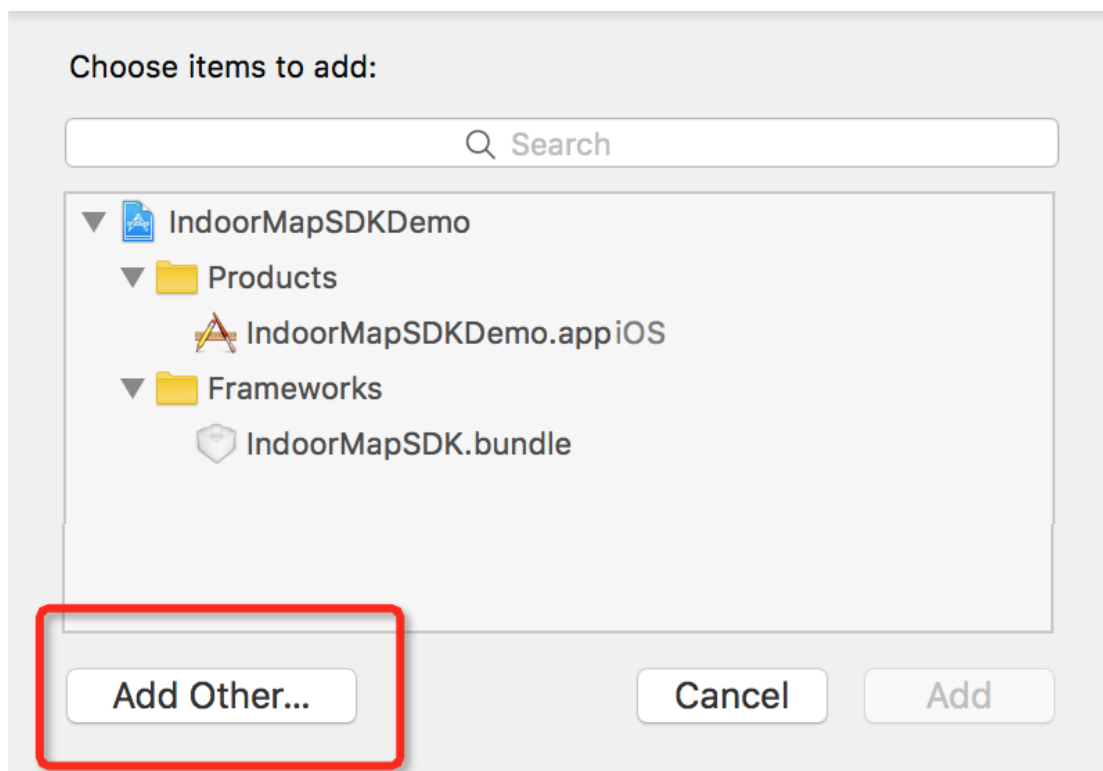
iOS 室内定位 SDK 的环境配置非常简单，只要引入室内定位库和相应的系统库即可。

左侧目录中选中工程名，在 TARGETS->General->Linked Frameworks and Libraries 中点击 “+” 按

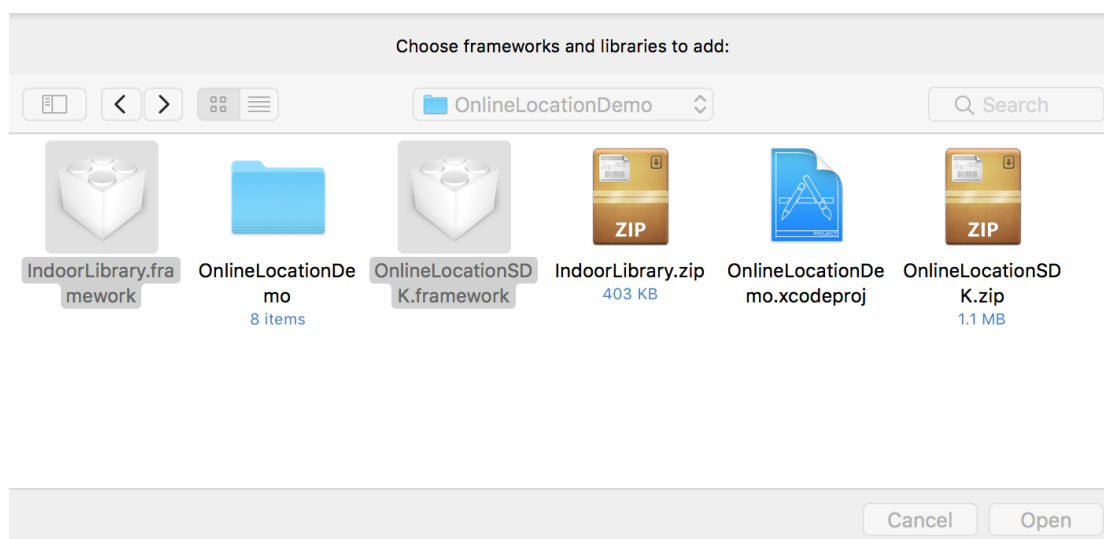


钮。

在弹出的窗口中点击 “Add Other” 按钮，选择解压后的 IndoorLibrary.framework 和 OnlineLoca-












tion.framework 文件添加到工程中即可。



完成 OnlineLocation.framework 文件的添加，再添加下图中所有其它的系统库，即完成了环境配置。

▼ Linked Frameworks and Libraries

Name	Status
 libz.tbd	Required ⬆⬇⬆
 libc++.tbd	Required ⬆⬇⬆
 SystemConfiguration.framework	Required ⬆⬇⬆
 CoreBluetooth.framework	Required ⬆⬇⬆
 CoreLocation.framework	Required ⬆⬇⬆
 CoreMotion.framework	Required ⬆⬇⬆
 CoreTelephony.framework	Required ⬆⬇⬆
 IndoorLibrary.framework	Required ⬆⬇⬆
 OnlineLocationSDK.framework	Required ⬆⬇⬆
+ -	

## 4. 获取室内位置

### 4.2. 构造定位对象

示例：

```
OnlineLocation* indoorLocation = [[OnlineLocation alloc] initWithDelegate:self];
```

功能：

初始化室内定位模块。

方法：

```
-(instancetype)initWithDelegate:(id<IndoorLocationDelegate>)delegate;
```

参数：

参数	类型	说明
delegate	id<IndoorLocationDelegate>	接收室内定位事件

返回值：

类型	说明
OnlineLocation *	返回室内定位模块对象

## 4.2. 设置高德地图 API KEY

示例：

```
indoorLocation.key = @"您申请的高德地图 API 的 key";
```

功能：

设置高德地图 API 的 KEY，用于访问高德服务器鉴权。

属性：

```
@property(nonatomic, strong) NSString* key;
```

参数：

参数	类型	说明
key	NSString*	高德地图 API KEY

说明：

用户在使用本 SDK 之前需要获取高德地图 API Key，相关说明请参考高德 LBS 开放平台

<http://lbs.amap.com>。申请高德地图 API KEY 的具体流程请参考开放平台文档

<http://lbs.amap.com/api/ios-sdk/guide/key/>。

KEY 不正确将无法使用本 SDK 的各项功能。

### 4.3. 指定定位建筑

示例：

```
indoorLocation.buildingId = @"室内建筑 POIID ";
```

功能：

指定在哪栋建筑物内进行室内定位。

属性：

```
@property(nonatomic, strong) NSString* buildingId;
```

参数：

参数	类型	说明
buildingId	NSString*	建筑物的高德 POIID

说明：

必须指定建筑物，否则可能无法定位。

## 4.5. 启动定位

示例:

```
[indoorLocation startLocation];
```

功能：

启动室内定位模块，进行定位操作。

方法:

```
-(int)startLocation;
```

返回值：

类型	说明
int	0：启动室内定位成功 -1：室内定位已经启动

## 4.6. 停止定位

示例:

```
[indoorLocation stopLocation];
```

功能：

停止室内定位模块。

方法:

```
-(int)stopLocation;
```

返回值：

类型	说明
int	0：停止室内定位成功

	-1：室内定位没有启动
--	-------------

## 4.7. 获取版本号

示例：

```
[OnlineLocation getVersion];
```

功能：

获取定位模块的版本号，当前版本是 5.6.0

方法：

```
+(NSString*)getVersion;
```

返回值：

类型	说明
NSString*	定位模块的版本号

## 4.8. 定位成功事件

示例：

```
-(void)indoorLocation:(LocationBase*)indoorLocation didLocationResult:(LocationResult*)result
{
    self.result = result;
    NSString* text = [NSString stringWithFormat:@"定位结果：%@", self.result];
    [self.textView setText:text];
}
```

功能：

定位成功时获得定位结果

方法：

```
-(void)indoorLocation:(LocationBase*)indoorLocation
didLocationResult:(LocationResult*)result;
```

参数：

参数	类型	说明
indoorLocation	LocationBase *	定位模块实例
result	LocationResult*	定位结果

## 4.9. 定位失败事件

示例：

```
-(void)indoorLocation:(LocationBase*)indoorLocation
    didLocationError:(NSString*)buildingId error:(NSError*)error
{
    NSString* text = [NSString stringWithFormat:@"%@定位失败：%@", buildingId, error];
    [self.textView setText:text];
}
```

功能：

定位失败时获取失败原因

方法：

```
-(void)indoorLocation:(LocationBase*)indoorLocation
    didLocationError:(NSString*)buildingId
    error:(NSError*)error;
```

参数：

参数	类型	说明
----	----	----



indoorLocation	LocationBase*	定位模块实例
buildingId	NSString*	定位的建筑物 ID
error	NSError*	错误原因，error.code 是错误码

## 4.10. 定位结果

示例：

```
int floorId = result.floorId;
double longitude = result.longitude;
double latitude = result.latitude;
```

功能：

室内定位结果

结构：

```
@interface LocationResult : NSObject
@property(nonatomic) int type;
@property(nonatomic, strong) NSString* buildingId;
@property(nonatomic) double longitude;
@property(nonatomic) double latitude;
@property(nonatomic) int floorId;
@property(nonatomic) float accuracy;
@property(nonatomic) float orientation;
@end
```

字段：

参数	类型	说明
type	int	结果类型（0：ONLINE，1：OFFLINE）
buildingId	NSString*	定位的建筑物 ID
longitude	double	经度（-180~180）

latitude	double	纬度 ( -90~90 )
floorId	int	楼层 ( -100~-1 , 1-127 )
accuracy	float	精度 ( 单位米 )
orientation	float	方向 ( 0~360 )

## 5. 实用工具

### 5.1. 定位错误码

示例:

```
if (error.code == SCAN_ERROR_BLE_NOTALLOW)
```

功能:

室内定位失败的错误原因。

定义:

```
#define INDOOR_ERROR_CODE - 1 // 定位错误
#define SCAN_ERROR_BLE_UNSUPPORTED - 100 // 设备不支持蓝牙定位
#define SCAN_ERROR_BLE_NOTOPEN - 101 // 蓝牙没有打开
#define SCAN_ERROR_BLE_NOTALLOW - 102 // 定位权限被禁止
#define SCAN_ERROR_BLE_NORESULT - 103 // 蓝牙扫描没有结果
#define LOCATION_ERROR_IDNOTSUPPORT - 200 // 建筑物不支持定位
#define LOCATION_ERROR_NOFINGER - 201 // 没有指纹
#define LOCATION_ERROR_FEWFINGERS - 202 // 指纹太少
#define LOCATION_ERROR_ONLINE - 206 // 在线定位失败
#define LOCATION_ERROR_UUID - 207 // 没有蓝牙 UUID
#define NETWORK_ERROR_DISCONNECT - 300 // 网络连接失败
#define NETWORK_ERROR_MISMATCH - 301 // 网络类型不匹配
#define NETWORK_ERROR_CONNECTERROR - 302 // 网络连接路由错误
#define NETWORK_ERROR_TIMEOUT - 303 // 网络连接超时
#define NETWORK_ERROR_REQUEST - 304 // 网络访问错误
#define LBS_ERROR_REQUEST - 400 // LBS 请求错误
#define LBS_ERROR_KEY - 401 // LBS 的 KEY 错误
```

字段：

名称	值	说明
INDOOR_ERROR_CODE	-1	定位错误
SCAN_ERROR_BLE_UNSUPPORTED	-100	设备不支持蓝牙定位
SCAN_ERROR_BLE_NOTOPEN	-101	蓝牙没有打开或者无蓝牙
SCAN_ERROR_BLE_NOTALLOW	-102	定位权限被禁止
SCAN_ERROR_BLE_NORESULT	-103	蓝牙扫描没有结果
LOCATION_ERROR_IDNOTSUPPORT	-200	建筑物不支持定位
LOCATION_ERROR_NOFINGER	-201	没有指纹
LOCATION_ERROR_FEWFINGERS	-202	指纹太少
LOCATION_ERROR_ONLINE	-206	在线定位失败
LOCATION_ERROR_UUID	-207	没有蓝牙 UUID
NETWORK_ERROR_DISCONNECT	-300	网络连接失败
NETWORK_ERROR_MISMATCH	-301	网络类型不匹配
NETWORK_ERROR_CONNECTERROR	-302	网络连接路由错误
NETWORK_ERROR_TIMEOUT	-303	网络连接超时

NETWORK_ERROR_REQUEST	-304	网络访问错误
LBS_ERROR_REQUEST	-400	LBS 服务器请求错误
LBS_ERROR_KEY	-401	LBS 的 KEY 错误

说明：

LBS 服务器请求错误的具体原因可参见 NSError 的 domain 字段。

## 6. 示例中心

相关下载章节提供的 [iOS 高德 室内定位 SDK Demo 工程](#) 包含的示例均向您说明了如何在您的 iOS 应用中使用高德 室内定位 SDK。除此之外，您还可以在[开发指南](#)的每个章节中找到关键方法的代码段。

### 6.1. 高德室内定位 SDK 示例 Demo 工程

---

示例工程中包含很多个 `ViewController`，向您说明了 SDK 接口的常见用法。您可以直接导入到编译器中进行编译，查看演示。您也可以直接基于提供的示例代码为基础开发您的应用。

当您运行示例应用时，它会显示一个包含演示功能的列表，[开发指南](#)的每一个章节您都可以在这个列表中找到。您可以点击选择其中一项，在自己的设备上运行这些演示功能。

如果示例应用可以成功运行，但提示您 `INVALID_USER_KEY`，请确认您已经按照[入门指南](#)中所述的，向应用的配置文件中添加了高德 Key。

## 6.2. 开发指南中的代码段

开发指南的每个页面都提供了说明 SDK 接口功能的代码段，这些代码段均取自示例工程。例如，您可以参阅[室内定位](#)的实现指南，对应可运行的 `ViewController` 可在示例 Demo 的列表项“单次定位”中找到。

# 7. 示例代码

室内定位完整示例代码如下：

```
#import "ViewController.h"
#import <OnlineLocationSDK/OnlineLocationSDK.h> //引用室内定位模块头文件

@interface SampleIndoorViewController ()<IndoorLocationDelegate>

@property(n nonatomic, strong) OnlineLocation* indoor; //室内定位实例
@property(n nonatomic, strong) UIButton* button;
@property(n nonatomic, strong) UITextView* textView;
@property(n nonatomic, strong) LocationResult* result; //定位结果

@end

@implementation SampleIndoorViewController

-(void)viewDidLoad
{
    [super viewDidLoad];
    //页面布局（按钮和文本框）
    self.view.backgroundColor = [UIColor whiteColor];
    CGRect rect = self.view.frame;
    rect.size.height /= 2;
    self.button = [UIButton buttonWithType:UIButtonTypeSystem];
    self.button.frame = rect;
    [self.button setTitleColor:[UIColor blackColor] forState:UIControlStateNormal];
    [self.button.titleLabel setAdjustsFontSizeToFitWidth:YES];
    [self.view addSubview:self.button];
    [self.button addTarget:self action:@selector (onButtonClicked:)
                        forControlEvents:UIControlEventTouchUpInside];
    [self updateTitle];
    rect.origin.y += rect.size.height;
    self.textView = [[UITextView alloc] initWithFrame:rect];
    [self.view addSubview:self.textView];
}
```

```
}
//更新按钮上的文字
-(void)updateTitle
{
    NSString* title = self.indoor==nil?@"开始定位":@"停止定位";
    [self.button setTitle:title forState:UIControlStateNormal];
}
//更新文本框中的内容，显示定位结果
-(void)updateText
{
    NSString* text = [NSString stringWithFormat:@"定位结果: %@", self.re-
result];
    [self.textView setText:text];
}
//点击按钮，切换定位开关
-(void)onButtonClicked:(id)sender
{
    if (self.indoor==nil)
    {
        [self startIndoor];
    }
    else
    {
        [self stopIndoor];
    }
    [self updateTitle];
}
//启动室内定位
-(void)startIndoor
{
    if (self.indoor==nil)
    {
        self.indoor = [[OnlineLocation alloc] initWithDelegate:self];
        //获取定位实例
        self.indoor.key = @"0123456789abcdef0123456789abcdef";
        self.indoor.buildingId = @"室内建筑物 POIID"; //设置当前的建筑物，
如果不指定建筑物，程序需要先访问网络确定当前建筑物，因此在网络不通时不能定位
        [self.indoor startLocation]; //开始定位
    }
}
//停止室内定位
-(void)stopIndoor
{
    if (self.indoor!=nil)
    {
        [self.indoor stopLocation]; //停止室内定位
        self.indoor = nil;
    }
}
//关闭界面时停止定位
-(void)viewDidDisappear:(BOOL)animated
{
    [self stopIndoor];
}
//室内定位结果
-(void)indoorLocation:(LocationBase*)indoorLocation didLocationResult:(Location-
Result*)result
{
    self.result = result;
    NSString* text = [NSString stringWithFormat:@"定位结果: %@", self.re-
result];
}
```

```
        [self.textView setText:text];           //在文本框中显示定位结果
    }
    //室内定位失败
    -(void)indoorLocation:(LocationBase*)indoorLocation didLocationEr-
    ror:(NSString*)buildingId error:(NSError*)error
    {
        //error.code 中是定位失败的错误码
        NSString* text = [NSString stringWithFormat:@"%@@定位失败: %@", build-
        ingId, error];
        [self.textView setText:text];           //在文本框中显示失败消息
    }
@end
```