

高德地图 iOS 室内地图 SDK 开发指南

V2.2.4Beta 版

高德软件有限公司
2017 年 1 月·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

技术交流论坛：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

法律声明	1
目录1	
1. 概述.....	5
1.1. 室内地图 SDK.....	5
1.2. 面向的读者	5
1.3. 兼容性.....	5
1.4. 申请 API Key	6
1.5. 更新说明	6
1.5.1. SDK IOS V2.2.0 Beta 版	6
1.5.2. 室内地图 SDK IOS V2.1.0Beta 版.....	7
1.5.3. 室内地图 SDK IOS 2.0 版	9
2. 入门指南	10
2.1. 第 1 步：获取最新版本的 Xcode	10
2.2. 第 2 步：获取室内地图最新 SDK.....	10
2.3. 第 4 步：获取高德 Key	10
2.4. 第 5 步：Hello AMap	10
2.6. 第 6 步：连接 iOS 设备	11
2.7. 第 7 步：构建和运行您的应用	11
2.8. 后续步骤.....	12
3. 创建工程	13
3.1. 简介 13	
3.2. 获取 Key	13
3.2.1. 如何申请 Key	13
3.2.2. 如何获取 Bundle Identifier	14
3.3. 工程部署	15
3.3.1. 添加室内地图库	15
3.3.2. 添加系统库	17
3.3.3. 导入资源文件.....	17
3.4. 开发者注意事项	19
4. 创建地图	20
4.1. 下载地图	20
4.1.1. 开发者 KEY	20
4.1.2. 下载地图数据	20
4.1.2.1. 数据下载或更新成功	21
4.1.2.2. 数据下载或更新失败	21
4.1.3. 删除地图数据	22
4.1.4. 获取缓存建筑物列表	23
4.2. 显示地图	24
4.2.1. 简介	24

4.2.2.	开发者 Key	24
4.2.3.	建筑物 Id	24
4.2.4.	楼层显示	25
4.2.5.	设置建筑物及楼层	25
4.2.6.	加载室内地图事件	26
4.2.6.1.	开始加载地图事件	26
4.2.6.2.	地图加载完成事件	26
4.2.6.3.	地图加载失败事件	27
4.2.6.4.	开始切换楼层事件	27
4.2.6.2.	切换楼层完成事件	28
4.2.6.3.	切换楼层失败事件	28
4.3.	周边数据	29
4.3.1.	显示周边数据开关	29
4.3.2.	加载周边数据	29
5.	与地图交互	31
5.1.	控件交互	31
5.1.1.	比例尺	31
5.1.2.	指南针	31
5.1.3.	楼层控件	32
5.1.4.	缩放控件	32
5.2.	手势交互	33
5.2.1.	移动手势	33
5.2.2.	缩放手势	33
5.2.3.	旋转手势	33
5.2.4.	倾斜手势	34
5.2.5.	点击手势	34
5.3.	事件交互	35
5.3.1.	移动地图	35
5.3.2.	旋转地图	35
5.3.3.	缩放地图	36
5.3.4.	当前地图的比例	37
5.3.5.	放大	37
5.3.6.	缩小	37
5.3.7.	倾斜地图	38
5.3.8.	地图重置	38
5.3.8.1.	重置地图	38
5.3.8.2.	重置地图位置	39
5.3.8.3.	重置地图方向	39
5.3.9.	相关设置	40
5.3.9.1.	设置 View 中心	40
5.3.9.2.	设置 POI 缩放比例	41
5.4.	室内位置	42
5.4.1.	当前定位点开关	42

5.4.2.	设置当前定位点	42
5.4.3.	当前的跟随模式	43
6.	在地图上绘制	44
6.1.	绘制点标记	44
6.1.1.	添加标注	44
6.1.2.	添加标注数组	44
6.1.3.	删除标注	45
6.1.4.	删除标注数组	45
6.1.5.	获取标注 View	45
6.1.6.	显示标注	46
6.1.6.	OIMAnnotation 类型	46
6.1.6.	OIMAnnotationView 类型	47
6.1.6.1.	annotation	47
6.1.6.2.	image	47
6.1.6.3.	centerOffset	47
6.1.6.	OIMPinAnnotationView 类型	48
6.2.	绘制线	48
6.2.1.	添加覆盖物	48
6.2.2.	添加覆盖物数组	48
6.2.3.	删除覆盖物	49
6.2.4.	删除覆盖物数组	49
6.2.5.	获取覆盖物对应的渲染	50
6.2.6.	标注点	50
6.2.7.	线	51
6.3.	绘制面	51
7.	室内路径规划	55
7.1.	开发者 KEY	55
7.2.	初始化	55
7.3.	点到点	56
7.4.	点到 POI	56
7.5.	POI 到 POI	57
7.6.	POI 到点	58
7.7.	路算结果	59
7.7.1.	路算成功	59
7.7.2.	路算失败	59
7.7.3.	路算结果开关	60
7.7.4.	设置当前路算结果	60
7.8.	显示状态	61
7.8.1.	设置 POI 显示状态	61
7.8.2.	清除显示状态	62
8.	室内搜索	63
8.1.	按关键字搜索	63
8.2.	按类型搜索	64
8.3.	获取所有类型	64
8.4.	按高德分类列表搜索	65

8.5.按 POIID 搜索	66
8.6.按距离搜索	66
9.实用工具	68
9.1.错误码对照表	68
10.室内地图常见问题	70
10.1. 常见问题列表	70
10.2. 室内地图篇	72
10.3. 室内定位篇	73
10.4. 数据生产篇	74
11.常用链接	77
11.1. 室内地图数据生产单位	77
11.2. 室内地图方案及案例	77
11.3. 室内地图 JS API	78
11.4. 室内地图 SDK 地址	78
11.5. 开发论坛及 QQ 群	78

1. 概述

1.1. 室内地图 SDK

高德地图 iOS 室内地图 SDK 是一套室内地图开发调用接口，供开发者在自己的 iOS 应用中加入室内地图相关的功能。开发者可以轻松地开发出室内地图显示与操作、兴趣点搜索、室内路线规划等功能。功能列表如下：

- SDK 主功能包括室内地图渲染、搜索、路算。
- 完美支持 iOS 手机、ipad 平板，可在不同屏幕尺寸下呈现完美的显示效果。
- 地图采用矢量方法绘制，使得地图处理速度更快、流量占用更少。
- 室内地图 3D 展现，效果更佳。
- 提供室内聚合分类搜索，快速便捷。
- 支持点到点的跨楼层室内路径规划，Step by Step 的路径引导。
- 地图显示支持自定义样式
- 支持外延数据的显示

1.2. 面向的读者

高德室内地图 iOS 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。

1.3. 兼容性

支持 iOS 7.0 及以上系统。

1.4. 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](#)。

1.5. 更新说明

1.5.1. SDK IOS V2.2.4 Beta 版

- ☐ 拆分为 IndoorLibrary 和 IndoorMapSDK
- ☐ 修复倾斜时的显示问题
- ☐ 开放楼层列表、指南针、比例尺等控件

1.5.2. SDK IOS V2.2.3 Beta 版

- ☐ 解决与高德地图 SDK 兼容问题

1.5.3. SDK IOS V2.2.2 Beta 版

- ☐ 服务端增加地图访问权限控制。开发者申请建筑物 ID 时必须提供 Key 以便开通访问权限
- ☐ 修复若干 bug，提升质量
- ☐ 按距离搜索时支持跨楼层搜索，方便开发者灵活设置

1.5.4. SDK IOS V2.2.0 Beta 版

- 支持在地图上叠加自定义覆盖物，包括点线面
- 可自由设置 POI 的颜色和图标，用于开发者不同业务状态的区分显示
- 自定义检索：提供室内所有 POI 分类，开发者可自由搭建索引聚合
- 可获取 POI 详细信息，包括名称、类型、楼层等
- 自定义地图 View 窗口，可随地图旋转而同步旋转



图 1-1 自定义面示例图

1.5.2. 室内地图 SDK IOS V2.1.0Beta 版

- 自定义渲染样式，构架更加绚丽的个性化地图显示



- 室内建筑周边信息展示，可应用于园区、商业圈等情况，拓展室内地图应用场景；



- 提供路算结果，用户可以据此完成更细致的引导来配合更复杂的场景。；
- 修复若干 bug，提升质量；

1.5.3. 室内地图 SDK IOS 2.0 版

- 采用 3D 渲染，支持地图倾斜。
- 地图操作添加了动画显示效果，用户体验更加顺畅平滑。
- 提高了渲染效率与性能，加快了大建筑物的加载速度。
- 重构了 SDK，增加了程序的稳定性。



图 1-2 室内地图 3D、倾斜显示效果

另外，需要注意的是：

- 采用矢量地图数据，使用 OpenGL ES2.0 绘制，支持 3D 效果展示。
- 室内数据搜索支持按关键字搜索，按分类搜索，按距离搜索。
- 室内路径规划(路算)支持点与点，点与 POI，POI 与 POI 之间的路径规划。
- 地图显示样式使用配置文件，用户可以根据需要自行定制。

2. 入门指南

2.1. 第 1 步：获取最新版本的 Xcode

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

2.2. 第 2 步：获取室内地图最新 SDK

请从 <http://lbs.amap.com/api/ios-indoor-sdk/download/> 下载最新的室内地图 SDK。

2.3. 第 4 步：获取高德 Key

请前往[高德开放平台控制台](#)申请 iOS Key。

2.4. 第 5 步：Hello AMap

1.配置高德 Key 到代码中

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

```
//室内地图对象
```

```
@property(nonatomic, strong)OIMMapView* imMapView;
```

```
//初始化室内地图
```

```
self.imMapView = [[OIMMapView alloc] initWithFrame:self.view.bounds];
```

```
//设置 KEY
```

```
self.imMapView.key = @"您的 Key";
```

2.地图展示

在 ViewController.m 文件相应的方法中进行地图初始化，初始化的步骤：

1)构造 MAMapView 对象;

2)将 MAMapView 添加到 Subview 中。

对于 **3D** 矢量地图, 在 **viewDidLoad** 方法中添加代码:

//室内地图对象

```
@property(nonatomic, strong)OIMapView* imMapView;
```

```
- (void)viewDidLoad
```

```
{
```

```
    [super viewDidLoad];
```

```
    //初始化室内地图
```

```
    self.imMapView = [[OIMapView alloc] initWithFrame:self.view.bounds];
```

```
    //设置 KEY
```

```
    self.imMapView.key = @"您的 Key";
```

```
    //设置建筑物 ID
```

```
    self.imMapView.buildingId = @"您的建筑物 Id";
```

```
    //显示室内地图
```

```
    [self.view addSubview:self.imMapView];
```

2.6. 第 6 步：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。

此外, 您也可以使用 **iOS** 模拟器来运行您的应用。**Xcode** 中自带了多种模拟器, 您可以选择一种模拟器来构建和运行您的应用。

2.7. 第 7 步：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。

您可以看到在您的项目里有地图展示。如果您未看到地图, 那请检查您是否联网, 如果您看到了地图, 但是不能使用 **POI** 搜索和路径规划等相关功能, 请确认是否添加了您的高德 **Key**。

2.8. 后续步骤

您可能想观摩一些示例代码。您可以在开发者指南中阅读更多有关导航的内容。

3. 创建工程

3.1. 简介

iOS 室内地图 SDK 的环境配置非常简单，只要引入室内地图库和相应的系统库，并导入资源文件即可。

3.2. 获取 Key

3.2.1. 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。

3.2.2. 如何获取 Bundle Identifier

以下为您提供两种获取 Bundle Identifier 的方法。

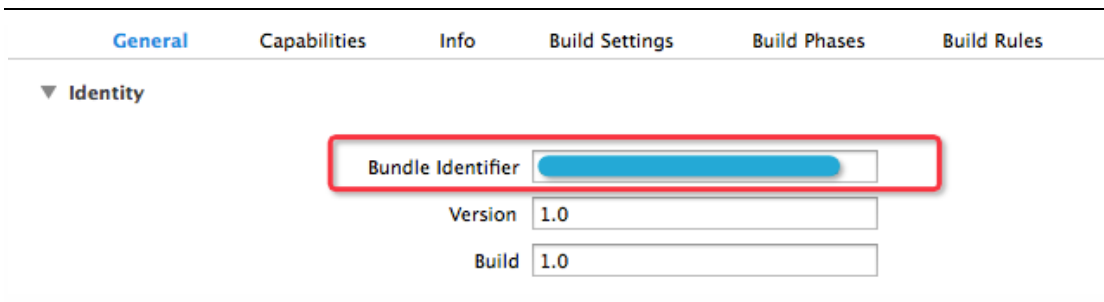
方法一

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：



3.3. 工程部署

3.3.1. 添加室内地图库

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

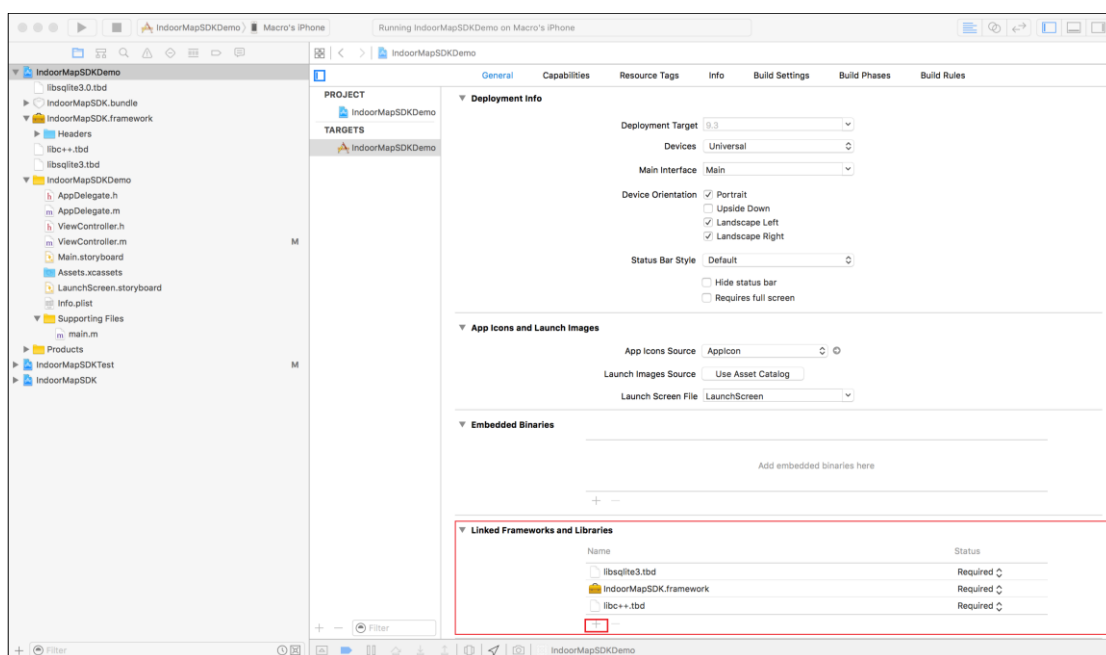


图 2-1 TARGETS->General->Linked Frameworks and Libraries

- 2、 在弹出的窗口中点击“Add Other”按钮，选择解压后的 IndoorLibrary.framework 和 Indoor-MapSDK.framework 文件添加到工程中即可。

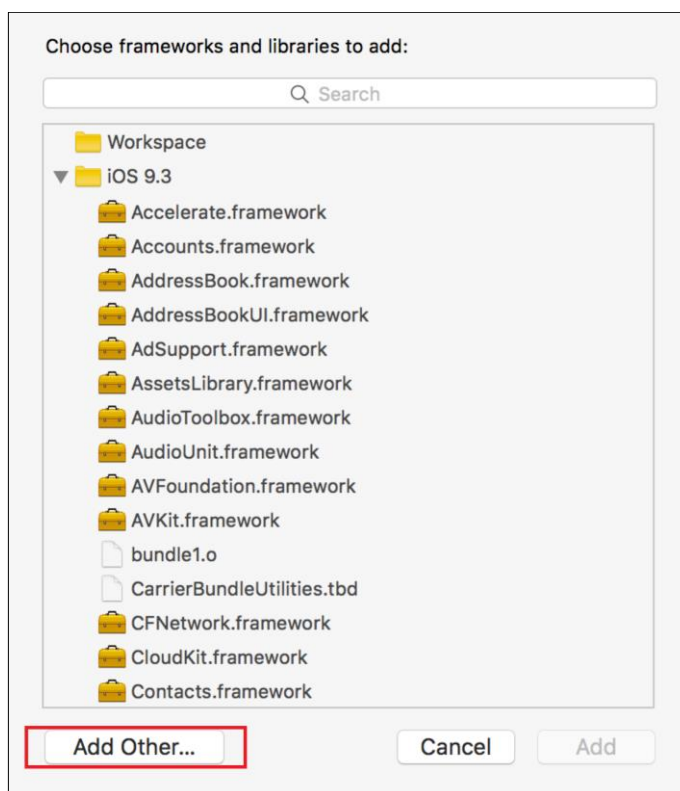


图 2-2 Add Other 按钮

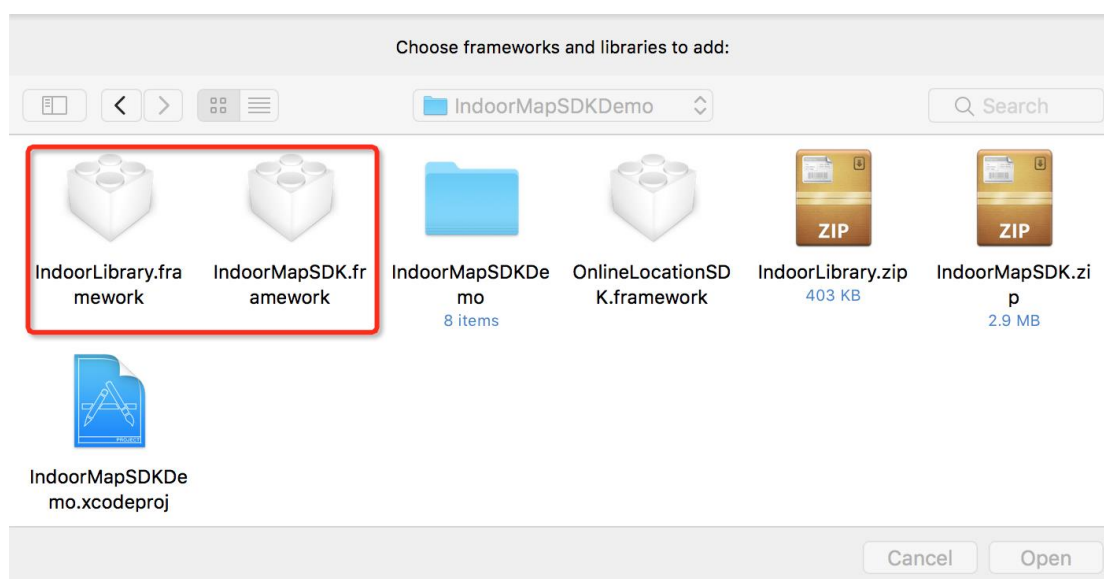


图 2-3 Add IndoorMapSDK.framework 文件

3.3.2. 添加系统库

完成 IndoorLibrary.framework 和 IndoorMapSDK.framework 文件的添加，再添加下图中所有其它的系统库。

▼ Linked Frameworks and Libraries

Name	Status
OnlineLocationSDK.framework	Required ⚙
IndoorLibrary.framework	Required ⚙
IndoorMapSDK.framework	Required ⚙
libc++.tbd	Required ⚙
libsqlite3.tbd	Required ⚙
libz.tbd	Required ⚙
CoreBluetooth.framework	Required ⚙
CoreLocation.framework	Required ⚙
CoreMotion.framework	Required ⚙
CoreTelephony.framework	Required ⚙
SystemConfiguration.framework	Required ⚙

图 2-4 添加其他系统库

3.3.3. 导入资源文件

下一步，导入资源文件

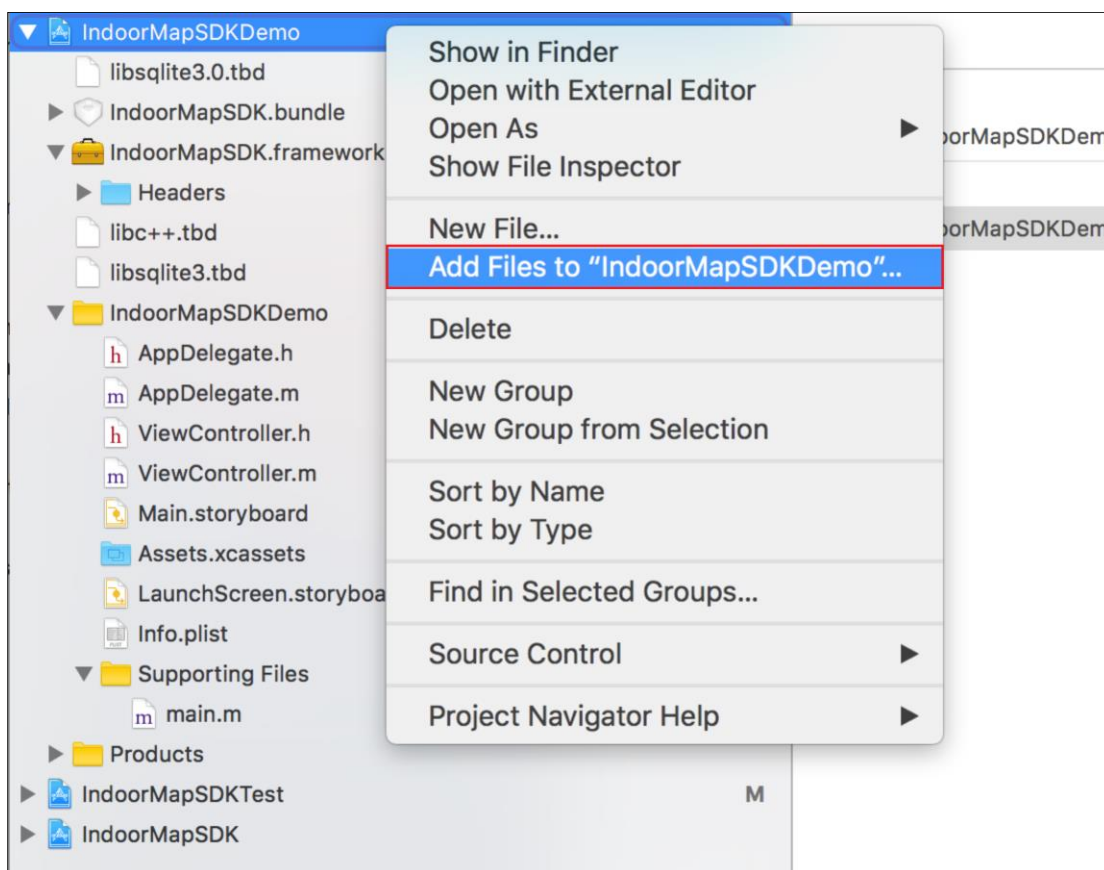


图 2-5 导入资源文件

选择 IndoorMapSDK.framework 中的 IndoorMapSDK.bundle。

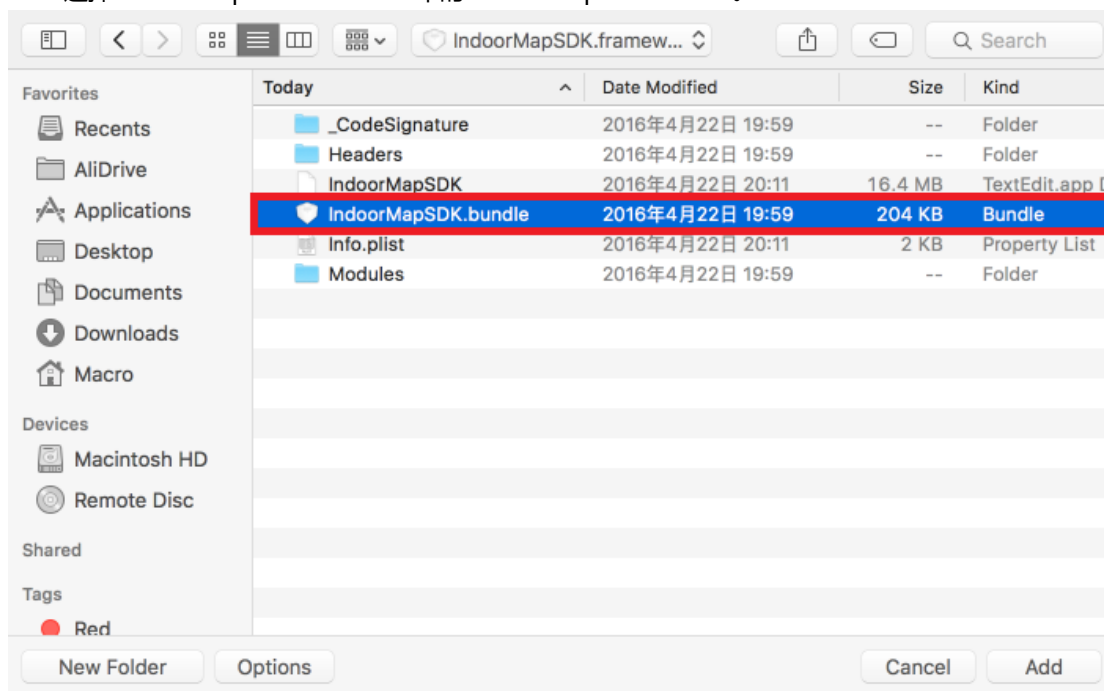


图 2-6 选择 IndoorMapSDK.bundle

按以上步骤即可成功导入资源文件即完成项目配置。如有疑问，请在官方 QQ 群及论坛社区联系我们的室内地图人员。

3.4. 开发者注意事项

注意：必须手工导入资源文件，否则地图无法正常加载

4. 创建地图

4.1. 下载地图

4.1.1. 开发者 KEY

函数：

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

说明：

KEY 的获取请到 <http://lbs.amap.com> 申请

如果 KEY 设置错误，将无法访问 LBS 服务器下载室内地图数据

请从 LBS 官网申请 KEY，并在程序中使用正确的 KEY

<http://lbs.amap.com/dev/key/>

4.1.2. 下载地图数据

函数：

```
-(int)downloadBuilding:(NSString*)buildingId;
```

参数说明：

参数	类型	说明
buildingId	NSString*	室内建筑物 Id

返回值：

返回值	类型	说明
IMOK	int	下载启动成功
IMERR	int	下载启动失败

说明：

- 1、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>
- 2、buildingId 必须是高德室内地图的建筑物 Id，否则无法成功下载数据
- 3、下载室内地图数据可以使用与地图显示不同的 buildingId，不影响地图显示。但不要与 OIM-MapView 使用同一个数据管理对象，因为 OIMMapView 会处理下载成功或失败的消息，可能会导致不可预知的错误

示例代码：

//下载室内地图数据

```
NSString* buildingId = @"请输入建筑物 ID";  
[self.imDataManager downloadBuilding:buildingId];
```

4.1.2.1. 数据下载或更新成功

函数：

```
-(void)dataManager:(OIMDataManager*)dataManager didDownloadSuccess:(NSString*)buildingId updated:(BOOL)updated;
```

参数说明：

参数	类型	说明
dataManager	OIMDataManager*	数据管理对象
buildingId	NSString*	室内建筑物 Id
updated	BOOL	是否有新数据下载下来

4.1.2.2. 数据下载或更新失败

函数：

```
-(void)dataManager:(OIMDataManager*)dataManager didDownloadFailure:(NSString*)buildingId error:(NSError*)error;
```

参数说明：

参数	类型	说明
dataManager	OIMDataManager*	数据管理对象

buildingId	NSString*	室内建筑物 Id
error	NSError*	失败相关的错误信息

4.1.3. 删除地图数据

1、删除指定建筑物的室内地图数据

函数：

```
-(int)removeBuilding:(NSString*)buildingId;
```

参数说明：

参数	类型	说明
buildingId	NSString*	室内建筑物 Id

返回值：

返回值	类型	说明
IMOK	Int	删除成功
IMERR	int	删除失败

说明：

1、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

2、该接口只会删除缓存中的数据文件，不影响已经加载到内存中的数据。也就是说删除 OIM-MapView 当前正在显示的建筑物数据，已经显示的地图不会消失

示例代码：

```
//删除室内地图数据
NSString* buildingId = @"请输入建筑物 ID";
[self.imDataManager removeBuilding:buildingId];
```

2、删除所有建筑物室内地图数据

函数：

```
-(int)removeAllBuildings;
```

返回值：

返回值	类型	说明
IMOK	int	删除成功
IMERR	int	删除失败

说明：

- 1、该接口会删除数据缓存路径下的所有文件，也就是所有的室内地图数据，请小心使用

示例代码：

```
//删除所有室内地图数据
[self.imDataManager removeAllBuildings];
```

4.1.4. 获取缓存建筑物列表

函数：

-(NSArray*)getAllBuildings;

返回值：

返回值	类型	说明
	NSArray*	所有已经缓存的建筑物列表

说明：

- 1、获取的是已经缓存的建筑物列表，而不是服务器上可供下载的建筑物列表，SDK 不提供可用的建筑物列表。

- 2、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

示例代码：

```
//获取室内建筑物列表
NSArray* buildingIds = [self.imDataManager getAllBuildings];
```

4.2. 显示地图

4.2.1. 简介

初始化地图模块，完成室内地图的加载

4.2.2. 开发者 Key

函数：

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

说明：

Key 的获取请到 <http://lbs.amap.com> 申请

如果 Key 设置错误，将无法访问 LBS 服务器下载室内地图数据

请从 LBS 官网申请 Key，并在程序中使用正确的 Key：<http://lbs.amap.com/dev/key/>

4.2.3. 建筑物 Id

函数：

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

说明：

请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

设置当前建筑物是异步操作，也就是说设置完成后立即获取，可能不是刚刚设置的新值

设置也有可能失败，失败后当前建筑物 Id 仍保持设置之前的值，在 OIMMapView 中使用会有相关事件。

示例代码：

//当前建筑物

```
NSString* buildingId = self.imMapView.buildingId;  
buildingId = @"请输入建筑物 ID";  
self.imMapView.buildingId = buildingId;  
//赋值后立即判断(如[self.imMapView.buildingId isEqualToString:buildingId]),  
结果可能为 NO
```

4.2.4. 楼层显示

函数：

```
@property(nonatomic) int floorNo;
```

说明：

设置当前楼层同样是异步操作，也有可能失败，在 OIMMapView 中使用会有相关事件

设置 floorNo 为 0，会加载当前建筑物的默认楼层，设置成功后 floorNo 不等于 0

示例代码：

```
//当前楼层
int floorNo = self.imMapView.floorNo;
floorNo = 2;
self.imMapView.floorNo = floorNo;
```

4.2.5. 设置建筑物及楼层

函数：

```
-(void)setBuildingId:(NSString *)buildingId floorNo:(int) floorNo;
```

参数	类型	说明
buildingId	NSString *	建筑物 Id
floorNo	int	楼层

参数说明：

说明：

1. 请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>
2. 该接口同时设置建筑物和楼层，[self.imMapView setBuildingId:@"请输入建筑物 ID" floorNo:n]; 与 self.imMapView.buildingId = @"请输入建筑物 ID";self.imMapView.floorNo = n;不完全等价，前者直接加载，后者先加载建筑物的默认楼层，再切换到指定楼层，除非 n 即是默认楼层

示例代码：

```
//当前建筑物
NSString* buildingId = @"请输入建筑物 ID";
int floorNo = 2;
[self.imMapView setBuildingId:buildingId floorNo:floorNo]; //直接加载
floorNo 这一层
//与下面的代码不完全等价
self.imMapView.buildingId = buildingId; //这一步会加载默认楼层
self.imMapView.floorNo = floorNo; //如果默认楼层不是 floorNo，再切换到
floorNo 这一层
```

4.2.6. 加载室内地图事件

打开一栋建筑物时，触发的是加载地图的相关事件，打开后再切换楼层，触发的是切换楼层相关的事件。

4.2.6.1. 开始加载地图事件

函数：

```
-(void)mapView:(OIMMapView*)mapView willStartLoading-  
Map:(NSString*)buildingId floorNo:(int)floorNo;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	将要加载的建筑物 Id
floorNo	int	将要加载的楼层

4.2.6.2. 地图加载完成事件

函数：

```
-(void)mapView:(OIMMapView*)mapView didFinishLoading-  
Map:(NSString*)buildingId floorNo:(int)floorNo;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	加载完成的建筑物 Id
floorNo	int	加载完成的楼层

4.2.6.3. 地图加载失败事件

函数：

```
-(void)mapView:(OIMMapView*)mapView didFailLoadingMap:(NSString*)buildingId floorNo:(int)floorNo withError:(NSError *)error;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	加载失败的建筑物 Id
floorNo	int	加载失败的楼层
error	NSError *	加载失败的错误信息

4.2.6.4. 开始切换楼层事件

函数：

```
-(void)mapView:(OIMMapView*)mapView willStartLoadingFloor:(NSString*)buildingId floorNo:(int)floorNo;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	将要加载的建筑物 Id
floorNo	int	将要加载的楼层

4.2.6.2. 切换楼层完成事件

函数：

```
-(void)mapView:(OIMMapView*)mapView didFinishLoad-  
ingFloor:(NSString*)buildingId floorNo:(int)floorNo;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	加载完成的建筑物 Id
floorNo	int	加载完成的楼层

4.2.6.3. 切换楼层失败事件

函数：

```
-(void)mapView:(OIMMapView*)mapView didFailLoad-  
ingFloor:(NSString*)buildingId floorNo:(int)floorNo withError:(NSError  
*)error;
```

参数说明：

参数	类型	说明
mapView	OIMMapView*	室内地图对象
buildingId	NSString*	加载失败的建筑物 Id
floorNo	int	加载失败的楼层
error	NSError *	加载失败的错误信息

4.3. 周边数据

4.3.1. 显示周边数据开关

函数：

```
@property(nonatomic) BOOL showExtension;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

4.3.2. 加载周边数据

函数：

```
-(int)loadExtensionData:(NSData*)extensionData;
```

参数：

参数	类型	说明
extensionData	NSData*	外延数据

返回值：

返回值	类型	说明
IMOK	int	成功

IMERR	int	失败
-------	-----	----

支持版本：2.1.0 及以上

5. 与地图交互

5.1. 控件交互

5.1.1. 比例尺

属性：

```
@property(nonatomic) BOOL showPlottingScale;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

属性：

```
@property(nonatomic, strong, getter=getMPlottingScaleView) UIImageView*
```

```
plottingScaleView;
```

5.1.2. 指南针

属性：

```
@property(nonatomic) BOOL showCompass;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

属性：

```
@property(n nonatomic, strong, getter=getMCompassView)UIImageView* compassView;
```

5.1.3. 楼层控件

函数：

```
@property(n nonatomic) BOOL showFloorList;
```

属性：

参数	类型	说明
YES		显示
NO		不显示

属性：

```
@property(n nonatomic, strong, getter=getMFloorView)UIView* floorView;
```

5.1.4. 缩放控件

属性：

```
@property(n nonatomic) BOOL showZoomControl;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

属性：

```
@property(n nonatomic, strong, getter=getMZoomInView)UIImageView* zoomInView;
```

```
@property(n nonatomic, strong, getter=getMZoomOutView)UIImageView* zoomOutView;
```

5.2. 手势交互

5.2.1. 移动手势

函数：

```
@property(nonatomic) BOOL enableTranslate;
```

取值：

参数	类型	说明
YES		允许
NO		不允许

5.2.2. 缩放手势

函数：

```
@property(nonatomic) BOOL enableScale;
```

取值：

参数	类型	说明
YES		允许
NO		不允许

5.2.3. 旋转手势

函数：

@property(nonatomic) BOOL enableRotate;

取值：

参数	类型	说明
YES		允许
NO		不允许

5.2.4. 倾斜手势

函数：

@property(nonatomic) BOOL enableIncline;

取值：

参数	类型	说明
YES		允许
NO		不允许

5.2.5. 点击手势

函数：

@property(nonatomic) BOOL enableClick;

取值：

参数	类型	说明
YES		允许
NO		不允许

5.3. 事件交互

5.3.1. 移动地图

函数：

```
-(int)updateTranslation:(float)x y:(float)y;
```

参数说明：

参数	类型	说明
x	float	X 轴方向的相对移动距离（单位像素），正值向右，负值向左
y	float	Y 轴方向的相对移动距离（单位像素），正值向上，负值向下

返回值：

返回值	类型	说明
IMOK	Int	成功
IMERR	int	失败

5.3.2. 旋转地图

函数：

```
-(int)updateRotation:(float)radian;
```

参数说明：

参数	类型	说明
----	----	----

radian	float	旋转的角度（单位度[-180,180]，正值逆时针，负值顺时针）
--------	-------	----------------------------------

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.3.3. 缩放地图

函数：

```
-(int)updateScale:(float) scale;
```

参数说明：

参数	类型	说明
scale	float	缩放的倍数（大于 1 放大，小于 1 缩小）

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.3.4. 当前地图的比例

函数：

```
@property(nonatomic, getter=getScaleUnit, setter=setScaleUnit:) float  
scaleUnit;
```

说明：

每个屏幕像素对应的实际地图距离，单位米/像素，不同大小的建筑物在打开时的默认比例是不同的。

5.3.5. 放大

函数：

```
-(int) zoomIn;
```

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.3.6. 缩小

函数：

```
-(int) zoomOut;
```

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.3.7. 倾斜地图

函数：

-(int)updateIncline:(float)incline;

参数说明：

参数	类型	说明
incline	float	倾斜的角度（单位度[-60,60]，正值前倾，负值后仰）

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.3.8. 地图重置

5.3.8.1. 重置地图

函数：

-(int)resetMap;

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

说明：

地图的位置、旋转、缩放、倾斜等都重置为加载的默认状态
等价于 translation={0,0}, rotation=0, scale=1, incline=0。

5.3.8.2. 重置地图位置

函数：

- (int) resetMapCenter;

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

说明：

地图的中心点回到 View 的中心点，其它如旋转、缩放、倾斜都不变
等价于 translation={0,0}

5.3.8.3. 重置地图方向

函数：

- (int) resetMapDirection;

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

说明：

旋转和倾斜回到初始状态，其它如旋转、移动都不变，等价于 rotation=0, incline=0

5.3.9. 相关设置

5.3.9.1. 设置 View 中心

1、移动地图上的一点到 View 的中心

函数：

```
-(int)setCoordinateCenter:(float)longitude latitude:(float)latitude;
```

参数说明：

参数	类型	说明
longitude	float	地图经度
latitude	float	地图纬度

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

2、移动指定 POI 到 View 中心

函数：

```
-(int)setFeatureCenter:(OIMFeature*) feature;
```

参数说明：

参数	类型	说明
feature	OIMFeature*	指定的 POI

返回值：

返回值	类型	说明
-----	----	----

IMOK	int	成功
IMERR	int	失败

5.3.9.2. 设置 POI 缩放比例

说明：把指定的 POI 按合适的缩放比例显示到 View 中。

函数：

```
-(int)setFeaturesCenter:(NSArray*) features;
```

参数说明：

参数	类型	说明
features	NSArray*	指定的 POI 列表

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

5.4. 室内位置

5.4.1. 当前定位点开关

函数：

```
@property(nonatomic) BOOL showLocationPoint;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

5.4.2. 设置当前定位点

函数：

```
-(int)setLocationPoint:(OIMPoint)point orientation:(float)orientation  
accuracy:(float)accuracy;
```

参数说明：

参数	类型	说明
point	OIMPoint	定位点坐标，包括经纬度和楼层
orientation	float	定位的方向
accuracy	float	定位的精度

返回值：

返回值	类型	说明
IMOK	int	设置成功

IMERR	int	设置失败
-------	-----	------

5.4.3. 当前的跟随模式

函数：

```
@property(nonatomic) int followingMode;
```

说明：

当前有 3 种模式

OIMFollowingMode_DontFollow 地图不跟随模式，默认方式，显示动画

OIMFollowingMode_FollowLocation 地图位置跟随模式，根据定位点的位置移动地图

OIMFollowingMode_FollowLocationAndDirection 地图位置和方向跟随模式，根据定位点的位置移动地图，同时根据定位给出的方向旋转地图。

6. 在地图上绘制

6.1. 绘制点标记

6.1.1. 添加标注

函数：

```
-(void)addAnnotation:(id<OIMAnnotation>) annotation;
```

参数说明：

参数	类型	说明
annotation	id<OIMAnnotation>	要添加的标注

6.1.2. 添加标注数组

函数：

```
-(void)addAnnotations:(NSArray*) annotations;
```

参数说明：

参数	类型	说明
annotations	NSArray*	要添加的标注数组

6.1.3. 删除标注

函数：

```
-(void) removeAnnotation: (id<OIMAnnotation>) annotation;
```

参数说明：

参数	类型	说明
annotation	id<OIMAnnotation>	要删除的标注

6.1.4. 删除标注数组

函数：

```
-(void) removeAnnotations: (NSArray*) annotations;
```

参数说明：

参数	类型	说明
annotations	NSArray*	要删除的标注数组

6.1.5. 获取标注 View

函数：

```
-(OIMAnnotationView *)viewForAnnotation: (id<OIMAnnotation>) annotation;
```

参数说明：

参数	类型	说明
annotation	id<OIMAnnotation>	标注

返回值：

类型	说明
OIMAnnotationView*	标注 View

6.1.6. 显示标注

函数：

```
-(void) showAnnotations:(NSArray *)annotations animated:(BOOL) animated;
```

参数说明：

参数	类型	说明
annotations	NSArray*	要显示的标注数组
animated	BOOL	动画显示

6.1.6. OIMAnnotation 类型

属性：

@property (nonatomic, readonly) OIMCoordinate coordinate;

说明：

标注的中心坐标，其值是楼层和经纬度。

6.1.6. OIMAnnotationView 类型

6.1.6.1. annotation

属性：

@property (nonatomic, strong) id<OIMAnnotation> annotation;

说明：

AnnotationView 对应的 Annotation

6.1.6.2. image

属性：

@property (nonatomic, strong) UIImage *image;

说明：

显示的图标控件

6.1.6.3. centerOffset

属性：

@property (nonatomic) CGPoint centerOffset;

说明：

图标显示位置与 annotation 坐标之间的偏差，单位是像素。

6.1.6. OIMPinAnnotationView 类型

属性：

@property (nonatomic) OIMPinAnnotationColor pinColor;

说明：

标注图标的显示类型，当前支持三种颜色的样式

OIMPinAnnotationColorRed

OIMPinAnnotationColorGreen,

OIMPinAnnotationColorPurple

6.2. 绘制线

6.2.1. 添加覆盖物

函数：

-(void)addOverlay:(OIMOverlay*)overlay;

参数说明：

参数	类型	说明
overlay	OIMOverlay*	要添加的 Overlay

6.2.2. 添加覆盖物数组

函数：

```
-(void)addOverlays:(NSArray*) overlays;
```

参数说明：

参数	类型	说明
overlays	NSArray*	要添加的 Overlay 数组

6.2.3. 删除覆盖物

函数：

```
-(void)removeOverlay:(OIOverlay*) overlay;
```

参数说明：

参数	类型	说明
overlay	OIOverlay*	要删除的 Overlay

6.2.4. 删除覆盖物数组

函数：

```
-(void)removeOverlays:(NSArray*) overlays;
```

参数说明：

参数	类型	说明
overlays	NSArray*	要删除的 Overlay 数组

6.2.5. 获取覆盖物对应的渲染

函数：

```
-(OIMOverlayRenderer *)rendererForOverlay:(OIMOverlay*)overlay;
```

参数说明：

参数	类型	说明
overlay	OIMOverlay*	Overlay

返回值：

类型	说明
OIMOverlayRenderer*	Overlay 对应的 Renderer

6.2.6. 标注点

函数：

```
@property (nonatomic, readonly) OIMCoordinate coordinate;
```

属性说明：

属性	类型	说明
coordinate	OIMCoordinate	点坐标，经纬度加楼层

6.2.7. 线

函数：

```
+(instancetype)polylineWithPoints:(OIMPoint *)points count:(NSUInteger)count;
```

参数说明：

参数	类型	说明
points	OIMPoint*	线的点数据
count	int	点的数量

返回值：

类型	说明
OIMPolyline*	线对象

6.3. 绘制面

函数：

```
+(instancetype)polygonWithPoints:(OIMPoint *)points count:(NSUInteger)count;
```

参数说明：

参数	类型	说明
points	OIMPoint*	面的点数据
count	int	点的数量

返回值：

类型	说明
OIMPolygon*	面对象

7. 室内路径规划

7.1. 开发者 KEY

函数：

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

说明：

1. 请从 LBS 官网申请 Key，并在程序中使用正确的 Key，申请路径
<http://lbs.amap.com/dev/key/>
2. 如果 KEY 设置错误，将无法访问 LBS 服务器请求室内路算。

7.2. 初始化

函数：

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

参数说明：

参数	类型	说明
delegate	Id<OIMRoutePlanningDelegate>	路算事件接口 OIMRoutePlanningDelegate

返回值：

返回值	类型	说明
	instancetype	室内路算对象

7.3. 点到点

函数：

```
-(int)requestRoutePlanning:(NSString *)buildingId fromPoint:(OIM-Point)from toPoint:(OIMPoint)to;
```

参数说明：

参数	类型	说明
buildingId	NSString *	室内建筑物 Id
from	OIMPoint	路算起点坐标
to	OIMPoint	路算终点坐标

返回值：

返回值	类型	说明
IMOK	int	路算发起成功
IMERR	int	路算发起失败

说明：

- 1、点坐标需要包括楼层和经纬度
- 2、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

7.4. 点到 POI

函数：

```
-(int)requestRoutePlanning:(NSString *)buildingId fromPoint:(OIM-Point)from toPoiId:(NSString*)to;
```

参数说明：

参数	类型	说明
buildingId	NSString *	室内建筑物 Id
from	OIMPoint	路算起点坐标
to	NSString*	路算终点 POIID

返回值：

返回值	类型	说明
IMOK	int	路算发起成功
IMERR	int	路算发起失败

说明：

- 1、点坐标需要包括楼层和经纬度
- 2、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

7.5. POI 到 POI

函数：

```
-(int)requestRoutePlanning:(NSString *)buildingId
fromPoiId:(NSString*)from toPoiId:(NSString*)to;
```

参数说明：

参数	类型	说明
buildingId	NSString *	室内建筑物 Id
from	NSString*	路算起点 POIID
to	NSString*	路算终点 POIID

返回值：

返回值	类型	说明
IMOK	int	路算发起成功
IMERR	int	路算发起失败

说明：

请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

7.6. POI 到点

函数：

```
-(int)requestRoutePlanning:(NSString *)buildingId
fromPoiId:(NSString*)from toPoint:(OIMPoint)to;
```

参数说明：

参数	类型	说明
buildingId	NSString *	室内建筑物 Id
from	NSString*	路算起点 POIID
to	OIMPoint	路算终点坐标

返回值：

返回值	类型	说明
IMOK	int	路算发起成功
IMERR	int	路算发起失败

说明：

- 1、点坐标需要包括楼层和经纬度

2、请在线申请建筑物数据，建筑物数据申请详情请参阅：<http://lbs.amap.com/console/apply/>

7.7. 路算结果

7.7.1. 路算成功

函数：

```
-(void)routePlanning:(OIMRoutePlanning*)routePlanning didRoutePlanning-  
Success:(NSString*)buildingId routeFeature:(OIMFeature*)routeFeature;
```

参数说明：

参数	类型	说明
routePlanning	OIMRoutePlanning*	执行路算的对象
buildingId	NSString*	路算对应的室内建筑物 Id
routeFeature	OIMFeature*	路算的结果

说明：

- 1、路算结果中包含了从服务器返回的 json 字符串，可用 routeFeature.properties["@json"]获取

示例代码：

//显示路算结果

```
-(void)routePlanning:(OIMRoutePlanning *)routePlanning  
    didRoutePlanningSuccess:(NSString *)buildingId  
    routeFeature:(OIMFeature *)routeFeature  
{  
    [self.imMapView setRoutePlanning:routeFeature];  
}
```

7.7.2. 路算失败

函数：

```
-(void)routePlanning:(OIMRoutePlanning*)routePlanning didRoutePlanning-  
Failure:(NSString *)buildingId error:(NSError*)error;
```

参数说明：

参数	类型	说明
routePlanning	OIMRoutePlanning*	执行路算的对象

buildingId	NSString *	路算对应的室内建筑物 Id
error	NSError*	失败的相关信息

7.7.3. 路算结果开关

函数：

```
@property(nonatomic) BOOL showRoutePlanning;
```

取值：

参数	类型	说明
YES		显示
NO		不显示

7.7.4. 设置当前路算结果

函数：

```
-(int)setRoutePlanning:(OIMFeature*) feature;
```

参数说明：

参数	类型	说明
feature	OIMFeature*	路算结果（设置空 nil 的路算结果会清除地图上已经显示的路算）

返回值：

返回值	类型	说明
-----	----	----

IMOK	int	设置成功
IMERR	int	设置失败

7.8. 显示状态

7.8.1. 设置 POI 显示状态

函数：

```
-(int)setFeature:(OIMFeature*)feature status:(OIMFeatureStatus)status;
```

参数说明：

参数	类型	说明
feature	OIMFeature*	指定的 POI
status	OIMFeatureStatus	显示状态（高亮、蓝色气泡、红色气泡、路算起点、路算终点）

返回值：

返回值	类型	说明
IMOK	int	成功
IMERR	int	失败

7.8.2. 清除显示状态

函数：

```
-(void)clearStatus:(OIMFeatureStatus)status;
```

参数说明：

参数	类型	说明
status	OIMFeatureStatus	要清除的显示状态

返回值：

返回值	类型	说明
IMOK	void	成功
IMERR	void	失败

说明：

- 1、显示的状态包括：高亮、蓝色气泡、红色气泡、路算起点、路算终点
- 2、该接口清除所有 POI 的指定状态（暂不支持清除指定 POI 的指定状态）

8. 室内搜索

8.1. 按关键字搜索

函数：

```
-(NSArray*) searchByKeywords: (NSString*) keywords;  
-(NSArray*) searchByKeywords: (NSString*) keywords floorNo: (int) floorNo;
```

参数说明：

参数	类型	说明
keywords	NSString*	要搜索的关键字

返回值：

返回值	类型	说明
	NSArray*	搜索出的 POI 列表

说明：

返回的是 OIMFeature 对象列表

示例代码：

//按关键字搜索

```
NSArray* features = [self.imDataManager searchByKeywords:@"请输入搜索关键字"  
];  
for(OIMFeature* feature in features)  
{  
    NSString* name = feature.name;  
}  
[self.imMapView setFeatures:features status:OIMFeatureStatus_Checked];
```

8.2. 按类型搜索

函数：

```
-(NSArray*)searchByType:(NSString*)type;  
-(NSArray*)searchByType:(NSString*)type floorNo:(int) floorNo;
```

参数说明：

参数	类型	说明
type	NSString*	要搜索的类型

返回值：

返回值	类型	说明
	NSArray*	搜索出的 POI 列表

说明：

- 1、返回的是 OIMFeature 对象列表。
- 2、搜索的类型值可以从 getAllSearchTypes 返回的类型列表中取得。
- 3、每个类型的具体名称和含义需要从其它途径获取。

示例代码：

//按类型搜索

```
NSArray* features = [self.imDataManager searchByType:@"请输入搜索类型"];  
for(OIMFeature* feature in features)  
{  
    NSString* name = feature.name;  
}  
[self.imMapView setFeatures:features status:OIMFeatureStatus_Checked];
```

8.3. 获取所有类型

函数：

```
-(NSArray*)getAllSearchTypes;
```

返回值：

返回值	类型	说明
	NSArray*	所有可以搜索的类型列表

说明：

返回的是 NSString*类型的数组

示例代码：

//获取可以搜索的类型

```
NSArray* types = [self.imDataManager getAllSearchTypes];
for(NSString* type in types)
{
    [self.imDataManager searchByType:type];
}
```

注意：不同建筑物的数据类型可能不同。

8.4. 按高德分类列表搜索

函数：

```
-(NSArray*) searchByCategories: (NSArray*) categories;
-(NSArray*) searchByCategories: (NSArray*) categories floorNo: (int) floorNo;
```

参数说明：

参数	类型	说明
categories	NSArray*	要搜索的高德分类列表

返回值：

返回值	类型	说明
	NSArray*	搜索出的 POI 列表

说明：

返回的是 OIMFeature 对象列表。

示例代码：

//按类型搜索

```
NSArray* categories = [NSArray arrayWithObjects:@(50800), @(981151), nil];
NSArray* features = [self.imDataManager searchByCategories:categories];
for(OIMFeature* feature in features)
{
    NSString* name = feature.name;
}
[self.imMapView setFeatures:features status:OIMFeatureStatus_Checked];
```

8.5. 按 POIID 搜索

函数：

```
-(OIMFeature*) searchById: (NSString*) pid;
```

参数说明：

参数	类型	说明
pid	NSString*	要搜索的 id

返回值：

返回值	类型	说明
	OIMFeature*	搜索出的 POI 对象

说明：

返回的是 OIMFeature 对象

示例代码：

//按关键字搜索

```
OIMFeature* feature = [self.imDataManager searchById:@"请输入搜索 pid"];
```

8.6. 按距离搜索

函数：

```
-(NSArray*) search: (OIMPoint) position byDistance: (float) distance
limit: (int) limit;
-(NSArray*) search: (OIMPoint) position byDistance: (float) distance
byCategories: (NSArray*) categories limit: (int) limit;
```

```

- (NSArray*)searchAround:(NSString*)pid byDistance:(float)distance
limit:(int)limit;
- (NSArray*)searchAround:(NSString*)pid byDistance:(float)distance
byCategories:(NSArray*)categories limit:(int)limit;

```

参数说明：

参数	类型	说明
position	OIMPoint	指定的位置，包括经纬度和楼层
distance	float	离指定位置的距离（单位：米）
limit	int	POI 个数限制

返回值：

返回值	类型	说明
	NSArray*	搜索出的 POI 列表

说明：

- 1、返回的是 OIMFeature 对象列表
- 2、距离是按指定点到 POI 中心点的直线距离计算，并且只返回当前楼层的 POI
- 3、对象的 distance 属性值是指定点到 POI 中心点的直线距离，单位是米

示例：

```

//获取 10 米内最近的一个 POI
OIMPoint position = {120.2345, 39.6789, 1};
NSArray* features = [self.imDataManager search:position byDistance:10
limit:1];
for(OIMFeature* feature in features)
{
    NSString* name = feature.name;
    double distance = feature.properties["distance"];
}
[self.imMapView setFeatures:features status:OIMFeatureStatus_Checked];

```

9. 实用工具

9.1. 错误码对照表

增加室内地图 SDK 错误码对照表，样式参照如下：

当使用异步方法搜索信息时，回调接口将返回响应码。响应码列表如下：

	新版 响应 码	状态描述	问题排查策略
OIMError_Error	-1	错误（没有明确指定的错误）	错误（没有明确指定的错误），请提交工单，反馈给我们： http://lbs.amap.com/dev/ticket#/tickets?cid=7&create=true&tid=9002
OIMError_NetworkError	-300	网络错误（没有明确指定的网络错误）	网络错误（没有明确指定的网络错误）
OIMError_NetworkDisconnect	-301	网络连接错误（没有网络、网络不通、连接不上等错误）	网络连接错误（没有网络、网络不通、连接不上等错误）
OIMError_NetworkTimeout	-302	网络连接超时	网络连接超时
OIMError_ContentWrong	-303	网络返回的内容不可识别	网络返回的内容不可识别
OIMError_LBSError	-400	访问 LBS 服务器失败	访问 LBS 服务器失败

OIMError_LBSErrorKey	-401	访问 LBS 服务器时使用的 KEY 不对	访问 LBS 服务器时使用的 KEY 不对
OIMError_StyleError	-500	加载地图样式失败	加载地图样式失败
OIMError_FloorNoInvalid	-511	无效的或不存在的楼层	无效的或不存在的楼层
OIMError_RouteInvalid	-521	无效的路算结果	无效的路算结果
OIMError_DataError	-530	数据错误	数据错误
OIMError_DataNotExist	-531	数据不存在	数据不存在，也可能是没有权限访问，请访问如下链接： http://lbs.amap.com/cooperation/indoordata/?tab=1 申请室内地图数据

10. 室内地图常见问题

10.1. 常见问题列表

用户遇到问题时，可以在我们的官方 QQ 群或者论坛—室内地图模块提交相应的问题描述，我们的研发同学会积极给与沟通与解答。另外，常见问题整理如下表所示，供各位参考。

模块	问题描述	备注
室内地图	Q1：如何利用高德室内地图 SDK 搭建自己的应用？	
	Q2：高德室内地图 SDK 可以实现什么功能？	
	Q3：高德室内地图用户可以自定义地图还是需要连接网络才能使用	
	Q4：自己已有的室内地图数据（非使用高德生产工具生产的数据），如何能在高德地图上显示？	
	Q5：我想在微信公众号或 PC 网页里嵌入室内地图，目前是否支持？	
	Q6：我想在 PC 端里嵌入室内地图，且做业务二次开发，目前是否支持？	
室内	Q1：高德采用什么方式实现的室内定位？	
	Q2：高德室内定位提供哪几种模式？有何区别？	

定位	Q3：高德室内定位支持哪几个平台？	
	Q4：室内定位需要链接网络吗？	
	Q5：需要什么才能实现室内定位？	
	Q6：WIFI 或蓝牙需要如何部署？高德负责提供定位部署及施工工作么？	
	Q7：我有自己的定位数据，能在高德使用么？	
	Q8：高德提供的 iOS 室内定位 SDK，主要是基于什么技术实现的？	
	Q9：iOS 支持 WIFI 定位么？	
	Q10：用户需要开启哪些硬件设备才能支持 IOS 室内定位？	
	Q11：iOS 室内定位 SDK 里的 PDR，需要手机硬件支持吗？	
	Q12：Android 支持蓝牙或 WIFI 定位么？	
	Q13：用户需要开启哪些硬件设备才能支持 Android 室内定位？	
	Q14：Android 室内定位 SDK 里的 PDR，需要手机硬件支持吗？	
室内地图数据	Q1：室内地图数据如何申请？	
	Q2：如何获得数据生产资质？	
	Q3：申请数据生产资质的要求？	
	Q4：室内地图数据生产都需要哪些工具？	
	Q5：我就想自己简单画个公司的地图放进去用来测试，能做到吗？	
	Q6：室内地图数据如何更新？	
	Q7：如何知道哪些建筑物有室内定位功能？	
	Q8：若没有室内定位数据，如何实现室内定位功能？	
	Q9：谁负责采集室内定位数据？高德会参与吗？	

	Q10：室内定位数据如何更新？	
--	-----------------	--

10.2. 室内地图篇

Q1：如何利用高德室内地图 SDK 搭建自己的应用？

如果要在自己的 APP 中使用，直接在高德地图 LBS 官网，申请 Key，下载高德开放的室内 SDK（室内地图+室内定位），整合到自己的应用中。

Q2：高德室内地图 SDK 可以实现什么功能？

室内地图 SDK 包括室内地图渲染显示、室内搜索、室内路算等功能；可结合高德室内定位 SDK 使用，也可单独使用。

Q3：高德室内地图用户可以自定义地图还是需要连接网络才能使用？

目前不支持离线地图包。您在二次开发的时候，可以让网络连接后强制下载，然后就不用联网了。

Q4：自己已有的室内地图数据（非使用高德生产工具生产的数据），如何能在高德地图上显示？

可通过 ArcGIS 进行数据整理，然后通过外业生产工具进行数据重新生产。若此建筑物高德已经有了，则可通过编辑平台进行数据更新，无需再次生产。

Q5：我想在微信公众号或 PC 网页里嵌入室内地图，目前是否支持？

可以支持。高德室内 H5 产品以 URL 的方式，零开发，即可实现 PC、移动端应用全覆盖。可以通过简单设置请求参数实现不同的室内地图展现形态。

室内外一体化展现：[杭州城西银泰城](#)（以城西银泰城为例）

室内独立展现：[北京朝阳大悦城](#)（以朝阳大悦城为例）

Q6：我想在 PC 端里嵌入室内地图，且做业务二次开发，目前是否支持？

暂不支持业务二次开发，只提供既有 H5 产品形态；JavaScript API 产品正在规划中。

10.3. 室内定位篇

Q1: 高德采用什么方式实现的室内定位？

高德使用 WIFI，或蓝牙作为基础信息，配合 PDR（惯导），地磁等辅助数据在指纹算法的基础上独立研发实现的具有自主知识产权的室内混合定位技术。

Q2: 高德室内定位提供哪几种模式？有何区别？

高德室内定位提供离线定位和在线定位。

1) 离线定位可以用户预先指定建筑物下载定位数据，现场可不链接网络直接定位；

2) 在线定位无需预下载指纹，现场通过链接网络实时定位。

可根据项目需求采用不同的方式实现室内定位。

Q3: 高德定位提供哪几个平台定位？

高德提供 iOS 和 Android 两个平台定位。

Q4: 室内定位需要链接网络吗？

室内定位提供两种定位方式：

方式 1：离线定位，若用户事先下载好定位数据，在现场无需链接网络。则可实现定位；

方式 2：在线定位，定位时需要链接网络，实现室内定位。

Q5: 需要准备什么才能实现室内定位？

可自行选择蓝牙或 WIFI 定位方式，选择实现技术后，需要现场有 WIFI 部署或蓝牙部署，并通过高德提供的定位数据采集工具进行定位指纹数据采集，数据准备完成后上传到高德服务器则可使用 SDK 实现室内定位。

Q6: WIFI 或蓝牙需要如何部署？高德负责提供定位部署及施工工作么？

WIFI 部署直接沿用通用的网络连接部署即可实现室内定位；

高德只提供基础的部署方案和策略，不参与施工与成本评估。

Q7: 我有自己的定位数据，能在高德使用么？

不可以，高德的数据和规格是独立的，需要使用高德的生产工具进行定位数据采集与生产。

Q8: 高德提供的 iOS 室内定位 SDK，主要是基于什么技术实现的？

高德的 iOS 室内定位 SDK，基于 iBeacon、PDR 等多种室内定位技术综合定位，定位精度更高，应用场景更广。

Q9: iOS 支持 WIFI 定位么？

暂时不支持。

Q10: 用户需要开启哪些硬件设备才能支持 IOS 室内定位？

用户只需要打开蓝牙设备（无需链接任何蓝牙扫描结果）；当用户使用在线定位时需要同时保障网络通畅即可实现室内定位。

Q11: iOS 室内定位 SDK 里的 PDR，需要手机硬件支持吗？

是的，需要手机硬件带 PDR；如果手机不支持 PDR，则定位时，不会使用到 PDR。

10.4. 数据生产篇

Q1: Android 支持蓝牙或 WIFI 定位么？

高德提供的 Android 定位 SDK 支持 WIFI 及蓝牙定位功能。

Q2: 用户需要开启哪些硬件设备才能支持 Android 室内定位？

如使用的基于 WIFI 的 Android 室内定位，用户只需要开启 WIFI 即可（无需链接上 WIFI）；

如使用的基于蓝牙的 Android 室内定位，用户只需要开启蓝牙即可（无需链接任何蓝牙扫描结果）。

Q3: Android 室内定位 SDK 里的 PDR，需要手机硬件支持吗？

是的，需要手机硬件带 PDR；

如果手机不支持 PDR，则定位时，不会使用到 PDR。

Q4: 室内地图数据如何申请？

室内地图数据目前分为两种：

第一种：高德已覆盖数据。

第二种：高德未覆盖数据。

直接通过线上进行 [室内地图数据申请](#)。高德会在数据库中进行查找，并告知需求方结果。

若为高德已覆盖数据，高德会根据数据申请流程发放建筑物 ID，本 ID 将配合高德提供的室内 SDK/API 等进行地图展现、搜索、路算、定位功能。

若为高德未覆盖数据，我们会提供两种方案：

方案一：提供已经有高德数据生产资质公司的联系方式、委托对方进行生产。

注：高德仅提供具有生产资质的公司联系方式，关于商务，价格，版权等信息。高德均不承担任何法律责任

方案二：申请自主生产，高德团队将与需求方线下沟通、评估能力。并组织培训，符合要求可提供免费的数据生产工艺、数据生产工具。并有限的支持需求方进行数据生产。

Q5：如何获得数据生产资质？

通过填写申请表格（公司介绍、室内地图生产经验、数据生产规模、计划生产的建筑情况等信息），进行登记，高德定期会组织地图生产培训，经过培训并通过考试的公司。经过备案后即可获得数据生产资质。

Q6：申请数据生产资质的要求？

必须对 ArcGIS 等 GIS 软件能够熟练应用；如有生产过数据的经验者优先；能投入人力进行外业生产与内业制作；满足以上要求者可在培训评估后申请资质。

Q7：室内地图数据生产都需要哪些工具？

ArcGIS,室内地图生产工具，室内地图编辑平台。

Q8：我就想自己简单画个公司的地图放进去用来测试，能做到吗？

不能，需要进行数据生产，可通过邮件申请测试数据，我们会提供一个离你最近的商场或其他建筑物进行测试。

Q9：室内地图数据如何更新？

高德室内团队提供室内地图更新维护平台，简单易学，能够快速上手，直接访问 [室内地图更新平台](#) 进行资质申请即可，收到申请评估确认后，会下发数据给申请账号，直接线上编辑更新即可。

Q10: 如何知道哪些建筑物有室内定位功能？

以邮件的形式发送申请给高德室内地图开放邮箱 后，室内团队会回复确认，并提供需要的室内定位建筑物的信息。

Q11: 若没有室内定位数据，如何实现室内定位功能？

需要生产定位数据（WIFI 或蓝牙指纹数据），生产定位数据前，需要先保证有室内地图数据。

Q12: 谁负责采集室内定位数据？高德会参与吗？

推荐具备室内图数据生产资质的公司采集，连同室内地图数据一并生产。原则上，高德原则上不参与数据生产，但会根据情况，如果是热度较高的建筑物通过申请，我们可以考虑进行生产。

Q13: 室内定位数据如何更新？

高德室内团队会开放室内定位更新工具，可由用户自行完成定位数据更新，更新后，上传相关数据给高德室内团队即可。

11. 常用链接

11.1. 室内地图数据生产单位

数据生产单位	联系人	联系方式
上海图聚信息技术有限公司	赵云阳	13701542048
中航材导航技术（北京）有限公司	王林军	18610295069
广东引道地图信息技术有限公司	邹善军	18602028806

11.2. 室内地图方案及案例

- 1、高德 LBS 官网室内定位解决方案地址：<http://lbs.amap.com/getting-started/indoorlocation/>
- 2、高德 LBS 官网室内地图解决方案地址：<http://lbs.amap.com/getting-started/indoorintro/>
- 3、高德 LBS 官网室内数据解决方案地址：<http://lbs.amap.com/cooperation/indoordata/>
- 4、高德 LBS 官网室内地图行业案例地址：<http://lbs.amap.com/smart/indoorcases/>

11.3. 室内地图 JS API

1. 室内地图 JS API:
<http://lbs.amap.com/api/indoormap-js-api/summary/>
2. 室内外一体化 JS API (高德地图 JS API): <http://lbs.amap.com/api/javascript-api/guide/indoormap/>

11.4. 室内地图 SDK 地址

- 1、Android 室内地图 SDK : <http://lbs.amap.com/api/android-indoor-sdk/summary/>
- 2、Android 室内定位 SDK : <http://lbs.amap.com/api/android-indoorlocation-sdk/summary/>
- 3、iOS 室内地图 SDK: <http://lbs.amap.com/api/ios-indoor-sdk/summary/>
- 4、iOS 室内定位 SDK: <http://lbs.amap.com/api/ios-indoorlocation-sdk/summary/>
- 5、Android 室内外一体化 SDK (高德地图 SDK) :
<http://lbs.amap.com/api/android-sdk/guide/create-map/indoor/>
- 6、iOS 室内外一体化 SDK (高德地图 SDK) :
<http://lbs.amap.com/api/ios-sdk/guide/create-map/indoor/>

11.5. 开发论坛及 QQ 群

另外，高德开发者论坛：<http://lbsbbs.amap.com/portal.php> 有室内地图版块，欢迎在那里发言讨论。

高德室内地图开放 QQ 群是：[300265574](#)，欢迎加入，一起探讨室内地图应用。