



河北师范大学软件学院  
Software College of Hebei Normal University

# 百度地图实战



Android教研室



# 自我介绍

丁盟

qq: 2622885094





● 周边雷达介绍 

● 周边雷达使用

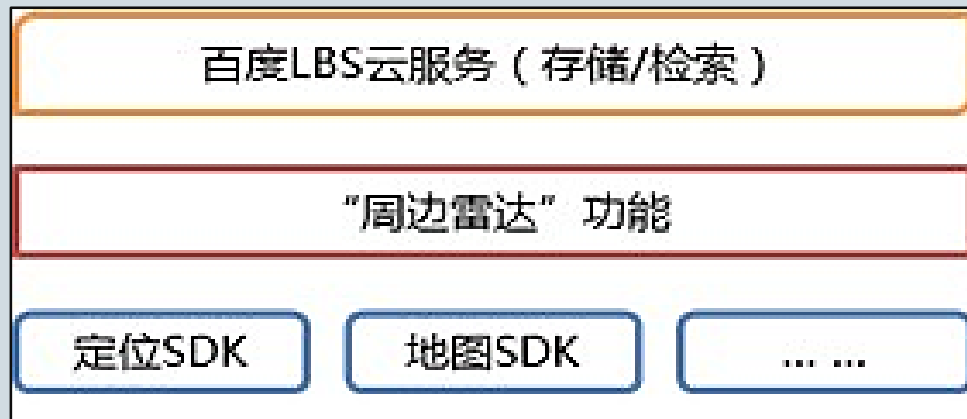




## 周边雷达介绍

### ❖ 什么是周边雷达功能？

周边雷达功能，是面向移动端开发者的一套SDK功能接口。同步支持Android和iOS端。它的本质是一个连接百度LBS开放平台前端SDK产品和后端LBS云的中间服务。





## 周边雷达介绍

### ❖ 利用周边雷达能够实现什么？

开发者利用周边雷达功能，可以便捷的在自己的应用内，帮助用户实现查找周边跟“我”使用同样一款App的人，这样一个功能。



## 周边雷达介绍

### ❖ 周边雷达功能的典型使用场景

1. 查看周边都有谁跟“我”使用同一个App，分布在哪里？
2. 查看周边用户在听什么歌、看什么文章、有什么新动态？
3. 查看周边有什么最新发生的新闻、资讯？



## 周边雷达介绍

### ❖ 使用周边雷达功能的流程

1. 注册周边雷达
2. 集成SDK
3. 开发



## 周边雷达介绍

### ❖ 注册周边雷达功能

在使用周边雷达功能之前，需要对应用的密钥（Key）做相应的注册操作。周边雷达支持应用位置信息上传，及一个或多个应用之间实现相互的位置信息查看功能。

注册周边雷达是使用其相应功能的基础前提。通过注册可实现一个或多个应用之间的关系绑定，实现相互之间的位置信息查看。





## 周边雷达介绍

访问如下地址，完成周边雷达的注册工作。

<http://lbsyun.baidu.com/index.php?title=radar>

我的周边雷达

新建周边雷达

1 填写描述名称

2 选择应用AK

3 确认注册

第一步：填写描述/名称：

\*该描述/名称会显示在“我的周边雷达”页面方便管理

放弃

下一步



## 周边雷达介绍

注册时选择需要启用周边雷达的App，然后选择完成。之后会在"我的周边雷达"中看到已经使用周边雷达的应用。

我的周边雷达

新建周边雷达

在使用周边雷达功能之前，需要对应用的密钥（Key）做相应的注册操作。

通过注册可实现一个或多个应用之间的关系绑定，实现相互之间的位置信息查看。

如下是您已建立的周边雷达注册关系。点击“删除”可解除周边雷达绑定关系，云端的数据也会被同步删除，请开发者慎重使用。

服务名称	应用ak	修改
测试Demo	MXXXXXXXXXXXXXXXXXXXXXlil	删除



● 周边雷达介绍 

● 周边雷达使用





## 周边雷达使用 - 初始化周边雷达功能

### ❖ 初始化周边雷达功能

在使用位置信息上传和检索周边位置信息之前，  
需要对周边雷达功能模块进行初始化操作。



## 周边雷达使用 - 初始化周边雷达功能

### ❖ Class **RadarSearchManager** - 周边雷达管理类。

初始化的核心代码如下：

```
// 获取周边雷达实例
RadarSearchManager mManager
    = RadarSearchManager.getInstance();
// 周边雷达设置监听，RadarSearchListener接口实现
mManager.addNearbyInfoListener(this);
// 周边雷达设置用户身份标识，id为null是设备标识，必须设置
mManager.setUserID(userID);
```



## 周边雷达使用 - 初始化周边雷达功能

static RadarSearchManager <b>getInstance()</b>	获取周边雷达实例
void <b>destroy()</b>	释放功能实例
void <b>addNearbyInfoListener</b> (RadarSearchListener listener)	添加查询周边信息的监听
void <b>clearUserInfo()</b>	清除当前使用用户的位置信息
boolean <b>nearbyInfoRequest</b> (RadarNearbySearchOption option)	查询周边的用户信息
void <b>removeNearbyInfoListener</b> (RadarSearchListener listener)	移除查询周边信息的监听
void <b>setUserID</b> (java.lang.String id)	设置用户ID
void <b>startUploadAuto</b> (RadarUploadInfoCallback callback, int internalSec)	启动自动上传自己的信息
void <b>stopUploadAuto()</b>	停止自动上传
boolean <b>uploadInfoRequest</b> (RadarUploadInfo info)	主动上传个人信息



## 周边雷达使用 - 接口

周边雷达功能模块在使用时还需实现如下两个接口：

### ❖ 接口 **RadarUploadInfoCallback**

- 上传信息回调接口。

PS：本接口在设置自动上传位置信息时使用。

- **RadarUploadInfo onUploadInfoCallback()**  
- 上传信息回调接口



## 周边雷达使用 - 接口

### ❖ 接口 **RadarSearchListener**

- 查询周边监听接口。

- void **onGetClearInfoState**(RadarSearchError error)  
- 清除位置信息监听
- void **onGetNearbyInfoList**(RadarNearbyResult result, RadarSearchError error)  
- 查询周边的人监听
- void **onGetUploadState**(RadarSearchError error)  
- 上传状态监听





## 周边雷达使用 - 位置信息上传

### ❖ 位置信息上传

周边雷达功能模块，支持将用户的位置等信息上传到百度LBS云服务，从而实现应用内部及应用之间的位置信息查看。

目前支持**单次位置信息上传**和**位置信息连续自动上传**两种模式。



## 周边雷达使用 - 位置信息上传

### ➤ 单次位置信息上传

```
//上传位置
```

```
RadarUploadInfo info = new RadarUploadInfo();  
info.comments = "用户备注信息";  
info.pt = pt;  
mManager.uploadInfoRequest(info);
```



## 周边雷达使用 - 位置信息上传

### ➤ 位置信息连续自动上传

```
// 设置自动上传的callback和时间间隔
mManager.startUploadAuto(this, 5000);
// 实现上传callback，自动上传
@Override
public RadarUploadInfo OnUploadInfoCallback() {
    RadarUploadInfo info = new RadarUploadInfo();
    info.comments = "用户备注信息";    // 备注信息
    info.pt = pt;                        // 坐标
    return info;
}
```



## 周边雷达使用 - 位置信息上传



### 上传结果监听

```
@Override
public void onGetUploadState(RadarSearchError error) {
    if (error == RadarSearchError.RADAR_NO_ERROR) {
        // 上传位置成功
    } else {
        // 上传位置失败
    }
}
```



## 周边雷达使用 - 清除用户信息

### ➤ 清除用户信息

```
mManager.clearUserInfo(); // 清空用户信息
// 处理清除的监听
@Override
public void onGetClearInfoState(RadarSearchError error) {
    if (error == RadarSearchError.RADAR_NO_ERROR) {
        // 清除位置成功
    } else {
        // 清除位置失败
    }
}
```



## 周边雷达使用 - 释放周边雷达资源

### ➤ 释放周边雷达资源

```
// 移除监听  
mManager.removeNearbyInfoListener(this);  
// 清除用户信息  
mManager.clearUserInfo();  
// 释放资源  
mManager.destroy();  
mManager = null;
```



## 周边雷达使用 - 周边位置检索

### ➤ 周边位置检索

利用周边雷达功能，可实现周边（处于同一个周边雷达关系内）用户位置信息检索的能力。检索过程支持距离、时间等约束条件；返回结果支持按照距离、时间远近的排序。



## 周边雷达使用 - 周边位置检索

### 发起周边位置检索

```
// 构造请求参数，其中centerPt是自己的位置坐标
RadarNearbySearchOption option
    = new RadarNearbySearchOption()
        .centerPt(pt)           // 搜索中心点
        .pageNum(00)           // 分页编号
        .pageCapacity(50)      // 每页容量
        .radius(2000);         // 检索半径

// 发起查询请求
mManager.nearbyInfoRequest(option);
```





## 周边雷达使用 - 周边位置检索

### 接收周边位置检索

```
@Override
public void onGetNearbyInfoList(
    RadarNearbyResult result,
    RadarSearchError error) {
    if(error==RadarSearchError.RADAR_NO_ERROR) {
        // 查询周边成功
        // result为周边信息结果
    } else {
        // 查询周边失败
    }
}
```



## 周边雷达使用

### ❖ 注意：

1. **mManager.setUserId()**必须进行设置，否则回调无反应；mManager.setUserId(null)为使用设备标识。
2. **info.comments**有长度限制，官方没有给出具体长度要求。



**Thank You!**

