**TZONE LBS SDK**

* **使用文档**

|  |  |
| --- | --- |
| **状态** | □ 草稿 □ 评审 √ 发布 □ 修订 |
| **版本号** | V1.1 |
| **作者** | Forrest wu | **发布时间** | 2016.01.14 |

# 前 言

 尊敬软件工程师，非常感谢您使用天圆产品“TZONE LBS”，

此文档为你介绍在Android平台下如何使用我们SDK快速开发应用（APP）。

目录

[前 言 2](#_Toc436382278)

[一、 概述 2](#_Toc436382279)

[1. SDK概述 2](#_Toc436382280)

[2. 功能概述 3](#_Toc436382281)

[二、 SDK开发准备 3](#_Toc436382282)

[1. 平台基本要求 3](#_Toc436382283)

[三、 SDK开发教程 3](#_Toc436382284)

[1. 如何使用Android Studio IDE引用 SDK 3](#_Toc436382285)

[2. SDK结构图。 6](#_Toc436382286)

[3. 添加权限。 6](#_Toc436382287)

[4. 功能类的使用。 7](#_Toc436382288)

[第一步，配置定位SDK参数 7](#_Toc436382289)

[第二步，实现LocationListener接口 7](#_Toc436382290)

[第三步，初始化LoactionServer 类 8](#_Toc436382291)

[第四步，开始定位 8](#_Toc436382292)

## 概述

### SDK概述

由于TZONE LBS是基于蓝牙BLE4.0开发出设备,所有目标平台必需支持BLE4.0，并且需要安装Android4.3系统平台。

### 功能概述

* 1. 室内定位

## SDK开发准备

### 平台基本要求

1. Android 4.3
2. 支持蓝牙BLE4.0

## SDK开发教程

### 如何使用Android Studio IDE引用 SDK

将“com.tzbeacon.sdk.jar、com.tzone.location.jar” 复制到项目libs文件夹下，如图：



右键属性”Add as Library”.



点击“ok”



### SDK结构图。



### 添加权限。

在工程的“ AndroidManifest.xml ”文件中进行添加，请直接拷贝。

<!-- 使用蓝牙设备的权限 -->

 <uses-permission android:name="android.permission.BLUETOOTH" />

 <!-- 管理蓝牙设备的权限 -->

 <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />

 <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />

 <!-- 只为具有BLE的设备提供 -->

 <uses-feature

 android:name="android.hardware.bluetooth\_le"

 android:required="true" />

 <!-- 往sdcard中写入数据的权限 -->

 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

 <!-- 在sdcard中创建/删除文件的权限 -->

 <uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS" />

 <!-- 禁止屏幕休眠和锁屏 -->

 <uses-permission android:name="android.permission.WAKE\_LOCK" />

 <!-- 网络访问权限 -->

 <uses-permission android:name="android.permission.INTERNET" />

### 功能类的使用。

#### 第一步，配置定位SDK参数

LocationOptions 类，设置定位参数包括：定位模式(LocationMode): 默认、高精度,区域信息、坐标比例（实际距离/坐标距离）、定位间隔等。

此处需要注意：LocationClient类必须在主线程中声明。需要Context类型的参数。 Context需要时全进程有效的context。

示例代码：

LocationOptions locationOptions = new LocationOptions(this

 , LocationOptions.LocationMode.Default //定位方式

 ,AreasList //区域信息

 ,BTSList //基站信息

 ,MapObj.BackgroundScale //坐标比例

 ,AppBase.Sys\_LocationInterval //定位间隔

 );

#### 第二步，实现LocationListener接口

LocationListener接口有2个方法需要实现： 1.接收异步返回的定位结果，参数是Location类型参数。2.接收异步返回的方向结果，参数是double 类型，从0~359分别北、东、南、西。

示例代码：

public class MyListener implements LocationListener{

 /\*\*

 \* 接收定位信息

 \* @param location 定位结果

 \*/

 @Override

 public void onReceiveLocation(Location location) {

 }

 /\*\*

\* 接收手机方向

\* @param direction

\*/

@Override

public void onReceiveSensorDirection(final int direction) {}

/\*\*

\* 手机是否在移动

\* @param isMove

\*/

@Override

public void onReceiveSensorIsMove(final boolean isMove) {}

 }

#### 第三步，初始化LoactionServer 类

示例代码:

LocationServer loactionServer = new(locationOptions,MyListener);

#### 第四步，开始定位

示例代码：

loactionServer.Start() //启动

start：启动定位SDK。 stop：关闭定位SDK。调用start之后只需要等待定位结果自动回调即可。

开发者定位场景如果是单次定位的场景，在收到定位结果之后直接调用stop函数即可。

如果stop之后仍然想进行定位，可以再次start等待定位结果回调即可。