**智慧图定位SDK-Android**

**V6.0.0**

**编写时间：** 2015-04-03

**北京智慧图科技有限责任公司**

目录

[1 概述 3](#_Toc415819231)

[1.1 定位SDK 3](#_Toc415819232)

[1.2 定位原理 3](#_Toc415819233)

[1.3 功能介绍 3](#_Toc415819234)

[2 开发指南 3](#_Toc415819235)

[2.1 简介 3](#_Toc415819236)

[2.1.1 面向的读者 3](#_Toc415819237)

[2.1.2 运行环境 4](#_Toc415819238)

[2.1.3 申请API Key 4](#_Toc415819239)

[2.2 V5.2 4](#_Toc415819240)

[2.2.1 导入库文件 4](#_Toc415819241)

[2.2.2 设置AndroidManifest.xml 5](#_Toc415819242)

[2.2.3 功能类说明 6](#_Toc415819243)

[2.3 V6.0 8](#_Toc415819244)

[2.3.1 导入库文件 8](#_Toc415819245)

[2.3.2 设置AndroidManifest.xml 9](#_Toc415819246)

[2.3.3 功能类说明 10](#_Toc415819247)

[3 示例 12](#_Toc415819248)

[附录 常见问题列表 13](#_Toc415819249)

# 概述

## 定位SDK

智慧图定位SDK-Android是为Android移动端应用提供一套简单易用的LBS定位服务接口，专注于为广大开发者提供更好的综合定位服务。通过使用智慧图定位SDK，开发者可以轻松为应用程序实现智能、精准、高效的定位功能，定位 SDK可单独使用，也可以结合智慧图地图 SDK 或其他地图 SDK。

## 定位原理



智慧图Android定位SDK是利用设备当前的Wi-Fi信息（[Wi-Fi定位](http://developer.baidu.com/map/index.php?title=android-locsdk/qa" \l ".E4.BB.80.E4.B9.88.E6.98.AFGPS.E5.AE.9A.E4.BD.8D.E3.80.81.E5.9F.BA.E7.AB.99.E5.AE.9A.E4.BD.8D.E5.92.8CWi-Fi.E5.AE.9A.E4.BD.8D.EF.BC.9F" \t "_blank)）和蓝牙信息（iBeacon定位）完成定位的。开发者在应用中成功集成智慧图定位SDK以后，即可以方便的通过定位SDK提供的接口向智慧图定位服务请求位置信息。

定位SDK会根据设备当前的实际情况（是否连接网络，是否扫描到Wi-Fi信息或beacon信息）生成定位依据进行定位。

## 功能介绍

混合定位：借助Wi-Fi、蓝牙和传感器信息，实现高精度的混合定位服务

# 开发指南

## 简介

### 面向的读者

本文档面向所有使用该SDK的开发人员、测试人员、合作伙伴及其他对此感兴趣的用户。

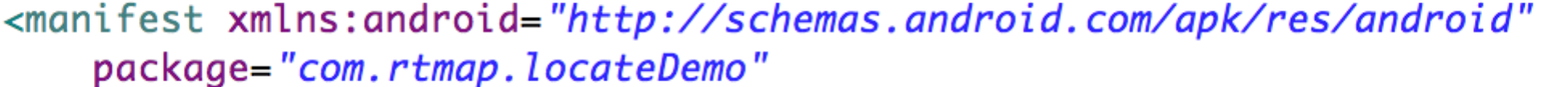
### 运行环境

|  |  |
| --- | --- |
| 定位模式 | 要求 |
| Wi-Fi定位 | Android2.2及以上 |
| 蓝牙定位（beacon） | Android4.3及以上，设备支持蓝牙4.0 |

### 申请API Key

为保证服务可以正常使用，您需与智慧图公司联系获取API Key，这个Key是您在我们的位置服务平台的唯一标识。

如申请API Key只做测试用只需提供APP包名，同时说明申请测试时长，包名是Android应用程序本身在AndroidManifest.xml 中定义的名称，例如：



申请正式API Key时还需提供合同号。

将APP包名、申请时长、合同号以以下格式发送至邮箱liuwenlong@rtmap.com。

主题：Rtmap API Key（定位SDK-Android）申请

正文：

APP包名：＊＊＊＊＊＊（必须）

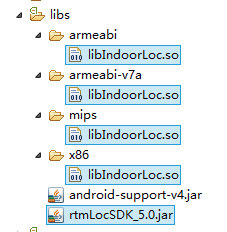
申请时长：＊＊＊＊＊＊（缺省默认一个月）

合同编号：＊＊＊＊＊＊（测试版可缺省）

## V5.2

### 导入库文件

在[相关下载](http://developer.baidu.com/map/index.php?title=android-locsdk/geosdk-android-download)最新的库文件。将so文件的压缩文件解压出来，把对应架构下的so文件放入开发者自己APP的对应架构下的文件夹中，建议全部放入，程序兼容性会大大提升，将rtmLocSDK\_x.x.jar文件拷贝到工程的libs目录下，这样您就可以在程序中使用智慧图定位SDK了。



### 设置AndroidManifest.xml

在application标签中声明service组件,每个app拥有自己单独的定位service

<service android:name=*"com.rtm.location.logic.UdpService"*

android:enabled=*"true"* >

</service>

声明使用权限

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

<uses-permission android:name=*"android.permission.CHANGE\_NETWORK\_STATE"* />

<uses-permission

android:name=*"android.permission.CHANGE\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission

android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />

<uses-permission

android:name=*"android.permission.INTERNET"* />

<uses-permission

android:name=*"android.permission.READ\_PHONE\_STATE"* />

<uses-permission

android:name=*"android.permission.BLUETOOTH"* />

<uses-permission

android:name=*"android.permission.BLUETOOTH\_ADMIN"* />

设置API Key

使用SDK需要在Mainfest.xml设置API Key（申请步骤详见上文），设置有误定位不能正常使用，在application标签中加入

<meta-data

android:name=*"RTMAP\_LOCATE\_ID"*//name值固定为RTMAP\_LOCATE\_ID，不可改变

android:value=*"12ABC3DEFJ"* /> //此处将12ABC3DEFJ替换为开发者申请的key

### 功能类说明

LocationApp是一个单例，必须在主线程中声明，需要Context类型的参数。Context需要全进程有效的context，推荐用getApplicationConext。

**private** MyBroadcastReceiver myBroadcastReceiver;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

// 定位服务初始化

LocationApp.*getInstance*().init(getApplicationContext());

// 注册广播，接收定位结果

myBroadcastReceiver = **new** MyBroadcastReceiver();

IntentFilter filter = **new** IntentFilter();

filter.addAction("BroadcastAction");

registerReceiver(myBroadcastReceiver, filter);

}

LocationApp是定位的SDK的核心类，具体方法如下：

初始化：

public void init(Context context); // 须在主线程中申明

设置定位服务器地址：

public void setServerAddress(String ip, String port);

启动定位：

public void start();

停止定位：

public void stop();

实现MyBroadcastReceiver，接收定位SDK返回信息：

**class** MyBroadcastReceiver **extends** BroadcastReceiver {

@Override

**public** **void** onReceive(Context context, Intent intent) {

String action = intent.getStringExtra("ACTION");

**if** (action.equals("RTMAP\_LICENSE")) {

Log.*i*(***TAG***, "License Check");

} **else** **if** (action.equals("RTMAP\_LOCATE")) {

ResultEntityParcelable output = intent

.getParcelableExtra("LOCATE");

ResultEntity locResult = output.getPosition();

StringBuffer sb = **new** StringBuffer();

// 定位时间戳

sb.append("time : ");

sb.append(locResult.getTimestampPDR());

sb.append("error code : ");

sb.append(locResult.getError());

sb.append("build id : ");

sb.append(locResult.getBuildID());

sb.append("floor id : ");

sb.append(locResult.getFloorID());

sb.append("indoor x : ");

sb.append(locResult.getCoordX());

sb.append("indoor y : ");

sb.append(locResult.getCoordY());

sb.append("indoor accuracy : ");

sb.append(locResult.accuracy);

Log.*i*(***TAG***, sb.toString());

}

}

}

ResultEntity类封装了定位SDK的定位结果，具体如下：

获取定位时间戳：

public long getTimestampPDR(); // 服务器定位时间，单位秒

获取定位错误码：

public int getError();

返回值：

|  |  |
| --- | --- |
| 错误码 | 错误信息 |
| 0 | 室内定位结果可用 |
| 1001、1 | 定位数据问题 |
| 1004、3 | 传感器数据问题 |
| 3018、3020、4 | 服务器忙 |
| 3017、5 | 无当前建筑物定位数据 |
| 601 | License验证失败 |
| 其他 | 服务器问题 |

获取建筑物ID：

public String getBuildID();

获取楼层号：

public int getFloorID();

楼层编码规则：

共5位

第1位：1代表地下(B)，2代表地上(F)

2-4位：代表楼层值

第5位：0代表无夹层，1代表夹层

例如：20120代表F12，20011代表F1.5，10010代表B1

获取定位坐标：

public int getCoordX(); // 单位毫米（地图左上角为坐标原点）

public int getCoordY(); // 单位毫米（地图左上角为坐标原点）

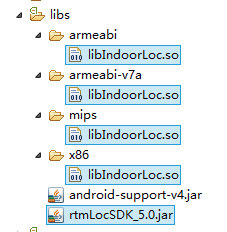
获取定位精度：

public int getAccuracy(); // 单位米

## V6.0

### 导入库文件

在[相关下载](http://developer.baidu.com/map/index.php?title=android-locsdk/geosdk-android-download)最新的库文件。将so文件的压缩文件解压出来，把对应架构下的so文件放入开发者自己APP的对应架构下的文件夹中，建议全部放入，程序兼容性会大大提升，将rtmLocSDK\_x.x.jar文件拷贝到工程的libs目录下，这样您就可以在程序中使用智慧图定位SDK了。



### 设置AndroidManifest.xml

在application标签中声明service组件,每个app拥有自己单独的定位service

<service android:name=*"com.rtm.location.logic.UdpService"*

android:enabled=*"true"* >

</service>

声明使用权限

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

<uses-permission android:name=*"android.permission.CHANGE\_NETWORK\_STATE"* />

<uses-permission

android:name=*"android.permission.CHANGE\_WIFI\_STATE"* />

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

<uses-permission

android:name=*"android.permission.ACCESS\_WIFI\_STATE"* />

<uses-permission

android:name=*"android.permission.INTERNET"* />

<uses-permission

android:name=*"android.permission.READ\_PHONE\_STATE"* />

<uses-permission

android:name=*"android.permission.BLUETOOTH"* />

<uses-permission

android:name=*"android.permission.BLUETOOTH\_ADMIN"* />

设置API Key

使用SDK需要在Mainfest.xml设置API Key（申请步骤详见上文），设置有误定位不能正常使用，在application标签中加入

<meta-data

android:name=*"RTMAP\_LOCATE\_ID"*//name值固定为RTMAP\_LOCATE\_ID，不可改变

android:value=*"12ABC3DEFJ"* /> //此处将12ABC3DEFJ替换为开发者申请的key

### 功能类说明

LocationApp是一个单例，必须在主线程中声明，需要Context类型的参数。Context需要全进程有效的context，推荐用getApplicationConext。

**private** MyBroadcastReceiver myBroadcastReceiver;

**public** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

// 定位服务初始化

LocationApp.*getInstance*().init(getApplicationContext());

// 注册广播，接收定位结果

myBroadcastReceiver = **new** MyBroadcastReceiver();

IntentFilter filter = **new** IntentFilter();

filter.addAction("BroadcastAction");

registerReceiver(myBroadcastReceiver, filter);

}

LocationApp是定位的SDK的核心类，具体方法如下：

获取SDK版本号：

public void getVersion();

获取so库版本号：

public void getLibsoVersion();

初始化：

public void init(Context context); // 须在主线程中申明

设置定位服务器地址：

public void setServerAddress(String ip, String port);

启动定位：

public void start();

停止定位：

public void stop();

实现MyBroadcastReceiver，接收定位SDK返回信息：

**class** MyBroadcastReceiver **extends** BroadcastReceiver {

@Override

**public** **void** onReceive(Context context, Intent intent) {

String action = intent.getStringExtra("ACTION");

**if** (action.equals("RTMAP\_LICENSE")) {

Log.*i*(***TAG***, "License Check");

} **else** **if** (action.equals("RTMAP\_LOCATE")) {

ResultEntityParcelable output = intent

.getParcelableExtra("LOCATE");

ResultEntity locResult = output.getPosition();

StringBuffer sb = **new** StringBuffer();

// 定位时间戳

sb.append("time : ");

sb.append(locResult.getTimestampPDR());

sb.append("error code : ");

sb.append(locResult.getError());

sb.append("build id : ");

sb.append(locResult.getBuildID());

sb.append("floor id : ");

sb.append(locResult.getFloorID());

sb.append("indoor x : ");

sb.append(locResult.getCoordX());

sb.append("indoor y : ");

sb.append(locResult.getCoordY());

sb.append("indoor accuracy : ");

sb.append(locResult.accuracy);

Log.*i*(***TAG***, sb.toString());

}

}

}

ResultEntity类封装了定位SDK的定位结果，具体如下：

获取定位时间戳：

public long getTimestampPDR(); // 服务器定位时间，单位秒

获取定位错误码：

public int getError();

返回值：

|  |  |
| --- | --- |
| 错误码 | 错误信息 |
| 0 | 室内定位结果可用 |
| 1001、1 | 定位数据问题 |
| 1004、3 | 传感器数据问题 |
| 3018、3020、4 | 服务器忙 |
| 3017、5 | 无当前建筑物定位数据 |
| 601 | License验证失败 |
| 其他 | 服务器问题 |

获取建筑物ID：

public String getBuildID();

获取楼层号：

public int getFloorID();

楼层编码规则：

共5位

第1位：1代表地下(B)，2代表地上(F)

2-4位：代表楼层值

第5位：0代表无夹层，1代表夹层

例如：20120代表F12，20011代表F1.5，10010代表B1

获取定位坐标：

public int getCoordX(); // 单位毫米（地图左上角为坐标原点）

public int getCoordY(); // 单位毫米（地图左上角为坐标原点）

获取定位精度：

public int getAccuracy(); // 单位米

# 示例

LocateDemo工程已经嵌入了SDK提供的基本功能接口，开发者只需要将API Key更换成自己申请的ID，然后运行，在可定位区域（定位数据已经采集，定位环境满足定位需要的区域）即可实现定位。

# 附录 常见问题列表

1. 智慧图定位SDK使用什么坐标系？

智慧图定位SDK使用坐标系以左上角为原点，X轴向右正向增长，Y轴向下正向增长，单位为毫米

1. 智慧图定位SDK的定位频率是多少，是否可调整？

定位频率默认为1s一次，此版本暂未提供定位频率设置接口

1. 为何我的程序无法定位？
2. 您需要确保程序编译通过。如果对自己写的代码有疑问，可以参照示例，查看标准源代码。
3. 请确保网络通畅，无论是连接wifi还是使用2G/3G/4G信号
4. 定位SDK的调用必须在主线程中
5. 请确保申请过智慧图API Key，并Manifest文件中正确设置
6. 请确保当前所处区域为室内定位覆盖区域（采集过WiFi指纹，或者现场布设有Beacon）
7. 请确保WiFi或者蓝牙处于开启状态
8. 请使用真机，在虚拟机上无法进行定位测试