**TZONE LBS SDK**

* **User Guide**

|  |  |
| --- | --- |
| **Status** | □ Draft □ Assess √ Release □ Edit |
| **Version** | V1.1 |
| **Author** | Forrest Wu | **Date** | 2016.01.14 |

# Preface

Distinguished software developer,

Thank you very much for using Tzone LBS. This user guide is to introduce how to develop App on Android platform quickly with our SDK.

Tzone Team

目录

[Preface 2](#_Toc440537359)

[1 Overview 2](#_Toc440537360)

[1.1 SDK overview 2](#_Toc440537361)

[1.2 Function overview 3](#_Toc440537362)

[2 Development preparation 3](#_Toc440537363)

[Platform basic requirement 3](#_Toc440537364)

[3 Development instructions 3](#_Toc440537365)

[3.1 Draw SDK into Android Studio IDE 3](#_Toc440537366)

[3.2 SDK structure 5](#_Toc440537367)

[3.3 Add authority 5](#_Toc440537368)

[3.4 Functional usage 6](#_Toc440537369)

[Step 1: Configure LBS SDK parameters 6](#_Toc440537370)

[Step 2: Realize LocationListener interface 6](#_Toc440537371)

[Step 3: Initialize LoactionServer type 7](#_Toc440537372)

[Step 4: Start positioning 7](#_Toc440537373)

## Overview

### SDK overview

As TZONE LBS was developed based on BLE 4.0, all the target platform should support BLE4.0, and install Android 4.3 system platform.

### Function overview

Indoor navigation

## Development preparation

### Platform basic requirement

1. Android 4.3
2. Supports BLE4.0

## Development instructions

### Draw SDK into Android Studio IDE

Input the “com.tzbeacon.sdk.jar, com.tzone.location.jar” into the libs folder of the project:



Right click Property ”Add as Library”.



Click “ok”



### SDK structure



### Add authority

Add in “ AndroidManifest.xml ” file of the project, please copy directly.

<!—permission to use Bluetooth device -->

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

 <!-- permission to manage Bluetooth device -->

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

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

 <!—only provide to device with BLE -->

 <uses-feature

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

 android:required="true" />

 <!—permission to write data into sdcard -->

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

 <!-- permission to create/delete file in sdcard -->

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

 <!—forbid screen sleep or lock -->

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

 <!—internet visiting permission -->

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

### Functional usage

#### Step 1: Configure LBS SDK parameters

For LocationOptions type, LBS parameter settings include: LocationMode: default, high accuracy, area info, coordinate scale (actual distance/coordinate distance), location distance.

Note here that LocationClient type must be declared in the main thread. Parameters of Context type are needed. The context should be valid in the whole process.

Example code:

LocationOptions locationOptions = new LocationOptions(this

 , LocationOptions.LocationMode.Default //Location mode

 ,AreasList //Area list

 ,BTSList //beacon list

 ,MapObj.BackgroundScale //Coordinate scale

 ,AppBase.Sys\_LocationInterval //Location interval

 );

#### Step 2: Realize LocationListener interface

There are two modes of LocationListener that needs realizing: 1. Receive location result returned by asynchrony, this type of parameter is Location type. 2. Receive direction result returned by asynchrony, this type of parameters is double type, it is north, east, south, west scaling 0~359.

Example code:

public class MyListener implements LocationListener{

 /\*\*

 \* Receive location info

 \* @param location Location result

 \*/

 @Override

 public void onReceiveLocation(Location location) {

 }

 /\*\*

\* Receive phone direction @param direction

\*/

@Override

public void onReceiveSensorDirection(final int direction) {}

/\*\*

\* Whether phone moving

\* @param isMove

\*/

@Override

public void onReceiveSensorIsMove(final boolean isMove) {}

 }

#### Step 3: Initialize LoactionServer type

Example code:

LocationServer loactionServer = new(locationOptions,MyListener);

#### Step 4: Start positioning

Example code:

loactionServer.Start() //Start

Start: start location SDK. Stop: stop location SDK. After calling start, only need to wait for location result callback automatically.

If the location scene of the developer is single location scene, then only need to call stop function directly after receiving location result.

If want to continue positioning after stop, then call start again and wait for location result callback.