

# 智石科技Beacon扫描及配置SDK for Android 集成指南

## 1、将BRTSDK添加到工程libs文件夹；

SDK下载: [https://github.com/BrightBeacon/BrightBeacon\\_Android\\_SDK](https://github.com/BrightBeacon/BrightBeacon_Android_SDK)

## 2、配置AndroidManifest.xml文件

```
<!-- 用于在线校验智石Beacon扫描SDK开发密钥 -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<!-- 蓝牙扫描权限,用于扫描蓝牙Beacon信标. -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

<!-- 安卓6.0及以后系统需要声明以下权限,才能扫描到蓝牙Beacon信标 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<!-- 如果APP编译targetSdk>=31(即安卓12及以上版本),需声明并在运行时申请以下权限用于扫描蓝牙Beacon和连接配置智石Beacon设备 -->
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
<!-- 如果APP编译targetSdk>=31(即安卓12及以上版本),使用SDK模拟广播蓝牙Beacon信号,需声明并在运行时申请以下权限 -->
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
```

安卓10及以上系统,请在中添加以下声明

```
<uses-library
    android:name="org.apache.http.legacy"
    android:required="false" />
```

## 3、编写BRTSDK集成代码

以下代码在Activity中添加:

```
private BRTBeaconManager beaconManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    //获取单例
```

```

beaconManager = BRTBeaconManager.getInstance(this);
// 注册应用 APPKEY申请:http://brtbeacon.com/main/index.shtml

beaconManager.registerApp("00000000000000000000000000000000");

// 通过设置 BRTBeaconManagerListener 接口实现，来检测Beacon的出现,显示以及更新。
BRTBeaconManager.setBRTBeaconManagerListener(beaconManagerListener);
}

// BRTBeaconManagerListener 接口实现。
private BRTBeaconManagerListener beaconManagerListener = new
BRTBeaconManagerListener() {

    @Override
    public void onUpdateBeacon(ArrayList<BRTBeacon> beacons) {
        // 扫描到的周边Beacon数据列表
    }

    @Override
    public void onNewBeacon(BRTBeacon beacon) {
        // 扫描到新的Beacon
    }

    @Override
    public void onGoneBeacon(BRTBeacon beacon) {
        // 扫描到的Beacon消失
    }
};

```

## 4、扫描周边Beacon

注意：

- 1、请确保测试手机蓝牙处理开启状态。
- 2、安卓6.0及以上系统扫描蓝牙Beacon信号，需要对以下权限添加运行时申请代码：

```

Manifest.permission.ACCESS_COARSE_LOCATION
Manifest.permission.ACCESS_FINE_LOCATION

```

- 3、安卓12.0及以上系统扫描蓝牙Beacon信号，需要对以下权限添加运行时申请代码：

```

Manifest.permission.BLUETOOTH_SCAN
Manifest.permission.BLUETOOTH_CONNECT
Manifest.permission.BLUETOOTH_ADVERTISE

```

- 4、执行以下代码控制蓝牙Beacon扫描和关闭：

```

// 开启Beacon扫描
BRTBeaconManager.startRanging();

// 关闭Beacon扫描
BRTBeaconManager.stopRanging();

```

提示：

回调函数是在非 UI 线程中运行的，请不要在回调函数中进行任何 UI 的相关操作，否则会导致 SDK 运行异常。如有需要，请通过 Handler 或者 Activity.runOnUiThread 方式来运行你的代码。

Beacon信息更新频率为 1 秒；发现一个新的Beacon后，如果在 8 秒内没有再次扫描到这个设备，则会回调onGoneBeacon。

## 5、监控Beacon进出状态

通常我们进入或离开某些Beacon设备时需要进行一些操作。下面是判断是否进入和离开MacAddress 为"000000000001"的Beacon的样例代码：

```
BRTBeaconManagerListener beaconManagerListener = new BRTBeaconManagerListener()
{

    @Override
    public void onUpdateBeacon(ArrayList<BRTBeacon> beacons) {
        // Beacon信息更新，每秒更新一次，内部设备列表默认有8秒缓存
    }

    @Override
    public void onNewBeacon(BRTBeacon beacon) {
        //新设备出现
        public void onNewBeacon(BRTBeacon beacon) {
            if (beacon.getMacAddress().equals("000000000001")){
                // 进入 MacAddress 为"000000000001 的Beacon
            }
        }

    @Override
    public void onGoneBeacon(BRTBeacon beacon) {
        //设备信号消失，默认有8s缓存
        if (beacon.getSerialNumber().equals("000000000001")){
            // 离开 MacAddress 为"000000000001 的Beacon
        }
    }
};

BRTBeaconManager.setBRTBeaconManagerListener(beaconManagerListener);
```

## 7、连接读取Beacon配置

如果我们要获取和配置Beacon的参数, 第一步需要连接进Beacon;

下面的代码片断需在Activity里面执行

```
BRTBeaconConnectionV2 conn = new BRTBeaconConnectionV2(this, null, beacon,
connectionListener);
```

```
BRTBeaconConnectionListener connectionListener = new BRTBeaconConnectionListener() {
```

```
    void onConnectedState(int newState, int status) {

        if (newState == BRTBeaconConnection.CONNECTED) {
```

```

        // 连接设备成功，可以读取配置参数；
        conn.readBeacon();
    } else {
        // 连接设备失败；
        conn.disconnect();
    }
}

void onBeaconRead(BRTBeacon beacon) {
    // Beacon配置读取完成；
}

void onBeaconWrite(BRTBeacon beacon, int status) {
    // Beacon配置更新完成；
}

void onError(BRTThrowable throwable) { }

void onCharacteristicChanged(String uuid, int status, byte[] value) { }

void onCharacteristicWrite(String uuid, int status, byte[] value) { }

void onCharacteristicRead(String uuid, int status, byte[] value) { }
}

```

## 8、更新Beacon参数

在连接Beacon成功以后,可以执行下面的代码更新Beacon参数;

```

BRTBeaconConfig config = new BRTBeaconConfig();
config.setName("BrTBeacon");
config.setMajor(1234);
config.setMinor(5678);
conn.writeBeacon(config);

```

如果设备参数更新完成, connectionListener 对象的 void onBeaconWrite(BRTBeacon beacon, int status)方法会被调用;

## 9、模拟广播Beacon信号

```

BRTAdvertiseData data = new BRTAdvertiseData(uuid, major, minor);
beaconManager.startAdvertising(data, advertiseListener);

```

