安卓6.0蓝牙权限开发

## 动态权限

在安卓6.0，谷歌调整了APP权限的分配方式，一些敏感的权限是在运行中动态分配给用户的，比如定位，相机等。而之前的系统是在安装apk时，系统就把权限列了出来，用户安装了apk就相当于同意了这个apk中所要求的所有权限。

另外，当编译时设置了targetSdkVersion >= 23，编译出来的包才需要6.0的系统中，需要处理动态权限分配，而targetSdkVersion<23时，则还是跟之前一样在安装apk时就拿到了所有权限。

在6.0中，扫描蓝牙需要拿到系统的定位权限，看起来似乎莫名其妙，但是谷歌确实把蓝牙权限划到了定位当中，因为蓝牙是可以测距的。

## 代码示例

需要在AndroidManifest.xml中申请android.permission.ACCESS\_COARSE\_LOCATION权限。

在进入到Activity时，需要判断蓝牙的权限

//判断是否有权限

if (ContextCompat.checkSelfPermission(this,

Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

//请求权限

  ActivityCompat.requestPermissions(this, new String[] { Manifest.permission.ACCESS\_COARSE\_LOCATION},

MY\_PERMISSIONS\_REQUEST\_ACCESS\_COARSE\_LOCATION);

//判断是否需要 向用户解释，为什么要申请该权限（用户点击了不再提醒）

if(ActivityCompat.shouldShowRequestPermissionRationale(this,

Manifest.permission.READ\_CONTACTS)) {

Toast.makeText(this, "shouldShowRequestPermissionRationale", Toast.LENGTH\_SHORT).show();

}

}

并复写Activity的onRequestPermissionsResult方法

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

// permissions是之前请求的权限列表，grantResults是请求结果,0为成功授权，其它为用户拒绝授权

}