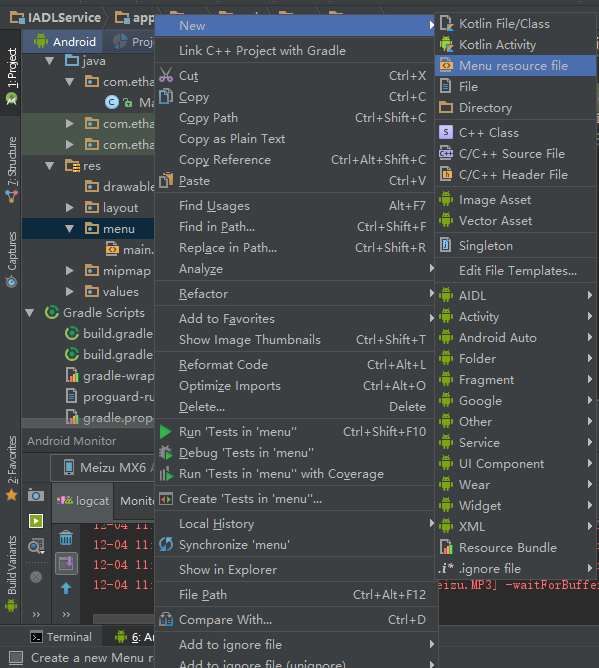
* gradlew是在Linux或mac系统中使用，gradlewbat是在windows系统中使用。
* 自定义过滤器

# 

* 项目中添加任何资源都会在R文件中生成一个相应的资源id
* 新建menu



* 启动Activity的最佳写法

在SecondActivity中

public static void actionStart(Context context, String data1, String data2) {  
 Intent intent = new Intent(context, SecondActivity.class);  
 intent.putExtra("params1", data1);  
 intent.putExtra("params2", data2);  
 context.startActivity(intent);  
}

在FirstActivity中

SecondActivity.*actionStart*(this, "data1", "data2");

* 禁止Button自动关机进行大写转换

android:textAllCaps="false"

* Nine-Patch(.9.png)

在上边框和左边框绘制 表示当前图片需要拉伸的区域

在下边框和右边框绘制 表示内容会被放置的区域

* Fragment回退

addTobackStack()，它可以接受一个名字用作描述返回栈的状态，一般出入null即可。

* App:popupTheme

设置菜单项的颜色

* CircleImageView GitHub地址

De.hdodenhof.circleimageview:2.1.0

* Snackbar参数一

需为CoordinatorLayout的子控件，否则无法监听到Snackbar的弹出和隐藏事件。

* android:fitsSystemWindows的作用

你的contentview是否忽略actionbar,title,屏幕的底部虚拟按键，将整个屏幕当作可用的空间。

<http://www.daxueit.com/article/9354.html>

* Alarm机制

从Android4.4系统开始，Alarm任务的触发时间将会变得不准确，有可能会延迟一段时间后任务才能得到执行。这并不是一个bug，而是系统在耗电性方面进行的优化。

**解决办法**

使用AlarmManager的setExact()方法来替代set()方法

* Doz e模式
  + - Alarm任务将会变得不准确

使用setAndAllowWhileIdle()或setExactAndAllowWhileIdle()，这两个方法之间的区别和set()、setExact()方法之间的区别是一样的。

* Git 合并分支

Git merge dev

* Fetch和pull的区别

Fetch同步代码不会合并到任何本地分支上，而是会存放到一个origin/master分支上，需要再调用merge命令将origin/master分支上的修改合并到主分支上。Pull相当于将这两条命令都执行了。

git fetch origin master

git merge origin/master

* 合并命令

Git merge dev

* 删除分支

Git branch -D dev

**有序广播截断**

abortBroadcast();

**本地广播**

使用LocalBroadcastManager对广播进行管理

**获取本地广播管理对象**

localBroadcastManager = LocalBroadcastManager.getInstance();

**注册本地广播监听器**

LocalReceiver localReceiver = newLocalReceiver();

localBroadcastManager.registerReceiver(localReceiver,intentFilter); //注册本地广播监听器

**发送本地广播**

localBroadcastManager.sendBroadcast(intent);

**取消注册本地广播**

localBroadcastManager.unregisterReceiver(localReceiver);

**杀死当前进程**

Android.os.Process.killProcess(android.os.Process.myPid());

killProcess只能用于杀掉当前程序的进程，不能使用这个方法去杀掉其他程序。

Context的OpenFileOutput / openFileInput

//所有的文件都默认存储到/data/data/包名/files/目录下  
  
//name: 不能包含路径  
//MODE\_PRIVATE 如果文件存在，会覆盖源文件的内容  
//MODE\_APPEND 如果文件存在，追加内容  
FileOutputStream os = openFileOutput("fileName", *MODE\_APPEND*);  
FileInputStream in = openFileInput("fileName");

**SharedPreferences apply和commit的区别**

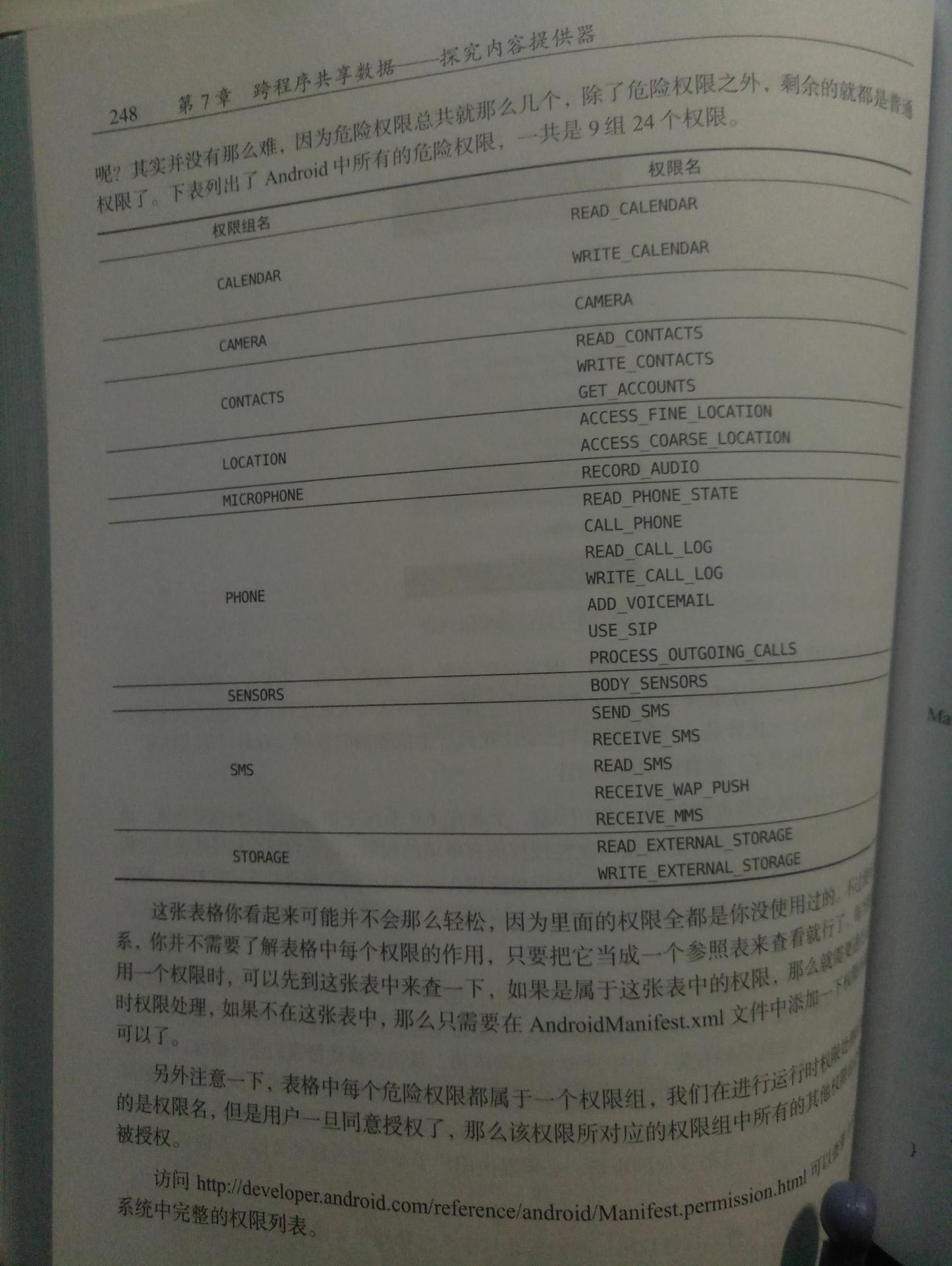
1. apply没有返回值而commit返回boolean表明修改是否提交成功
2. apply是将修改数据原子提交到内存, 而后异步真正提交到硬件磁盘, 而commit是同步的提交到硬件磁盘
3. apply方法不会提示任何失败的提示

apply的效率高一些，如果没有必要确认是否提交成功建议使用apply。

**SharedPreferences保存路径**

/data/data/包名/shared\_prefs/xxxxx

**危险的权限(需要申请的权限)**

****

**前台Service**

public class MyService extends Service {  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 Intent intent = new Intent(this, MainActivity.class);  
 PendingIntent pi = PendingIntent.*getActivity*(this, 0, intent, 0);  
 Notification notification = new NotificationCompat.Builder(this)  
 .setContentTitle("this is content title")  
 .setContentText("this is content text")  
 .setWhen(System.*currentTimeMillis*())  
 .setSmallIcon(R.mipmap.*ic\_launcher*)  
 .setLargeIcon(BitmapFactory.*decodeResource*(getResources(), R.mipmap.*ic\_launcher*))  
 .setContentIntent(pi)  
 .build();  
 startForeground(123456, notification);  
  
 }  
}