

## PendingIntent 和状态栏通知

### 1.1 知识解析

状态栏通知把一个图标添加到系统的状态栏（带有一个可选的文本提醒消息），并且还在通知窗口中添加一个通知消息。

当用户选择这个 PendingIntent 时, Android 发出这个通知所定义的一个 Intent（通常是启动一个 Activity）。

也可配置这个通知使用声音、震动和设备上的闪光灯来提醒用户。

一个活动或服务可以启动一个状态栏通知，然而活动只有在启动状态及处于焦点状态时才可以完成这个动作。

所以当用户在使用其他应用程序或设备处于休眠状态时，服务可以在背景中建立状态栏通知。

### 1.2 实战操作

使用 Status Bar Notification 的 4 个步骤：

第 1 步：获得 NotificationManager 对象

```
NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
```

第 2 步：获得 PendingIntent 对象

```
PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
intent, 0);
```

第 3 步：创建 Notification 对象

API11 后，使用 Notification.Builder 创建 Notification 对象：

```
Notification.Builder builder = new
```

```
Notification.Builder(mContext);
builder.set... (一系列 set 方法)
Notification notification = builder.build();//API 16 后 16 前用
getNotification()
```

API11 前:

```
(Notification notification =new Notification(R.drawable.icon,
“您有新消息了”, System.currentTimeMillis());)
```

第 4 步: 触发通知

```
notificationManager.notify(R.drawable.icon, notification);
```

修改 Status Bar Notification 的默认属性:

- notification.defaults = Notification.DEFAULT\_SOUND;// 使用默认的声音
- notification.defaults = Notification.DEFAULT\_VIBRATE;// 使用默认的震动
- notification.defaults = Notification.DEFAULT\_LIGHTS;// 使用默认的灯光
- notification.defaults = Notification.DEFAULT\_ALL;// 所有的都使用默认值

API 16 之后的写法:

```
Notification notify = new Notification.Builder(this)
setSmallIcon(R.drawable.message) // 设置状态栏中的小图片, 尺寸一般
建议在 24×24, 这个图片同样也是在下拉状态栏中所显示, 如果在那里需
要更换更大的图片, 可以使用 setLargeIcon(Bitmap icon)
setTicker(“TickerText:” + “您有新短消息, 请注意查收!”)// 设置在
status bar 上显示的提示文字
setContentTitle(“Notification Title”)// 设置在下拉 status bar 中
显示的标题
setContentText(“This is the notification message”)// status bar 中
显示的详细内容
setContentIntent(pendingIntent2) // 关联 PendingIntent, 点击后打开
的 Activity
build();// 需要注意 build() 是在 API level
// 16 及之后增加的, 在 API11 中可以使用 getNotificatin()
notify.flags |= Notification.FLAG_AUTO_CANCEL;
```

```
manager.notify(1, notify);
```

### Notification.flags 常用设置

- FLAG\_NO\_CLEAR: 不能被用户清除
- FLAG\_AUTO\_CANCEL: 用户在状态栏上点击该消息后被自动清除
- FLAG\_INSISTENT: 会一直响声直到用户打开状态栏
- FLAG\_ONGOING\_EVENT: 表示该 notification 是一个正在进行中的事件，如播放音乐、通话等
- FLAG\_SHOW\_LIGHTS: 亮 LED 灯或其他指示灯（如呼吸灯）

## 1.3 功能演示



## 1.4 职业素质

pendingIntent 是一种特殊的 Intent。主要的区别在于 Intent 的执行立刻的，而 pendingIntent 的执行不是立刻的。pendingIntent 执行的操作实质上是参数传进来的

Intent 的操作，但是使用 `pendingIntent` 的目的在于它所包含的 Intent 的操作的执行是需要满足某些条件的。

主要的使用的地方和例子：通知 `Notification` 的发送，短消息 `SmsManager` 的发送和警报器 `AlarmManager` 的执行等等。