

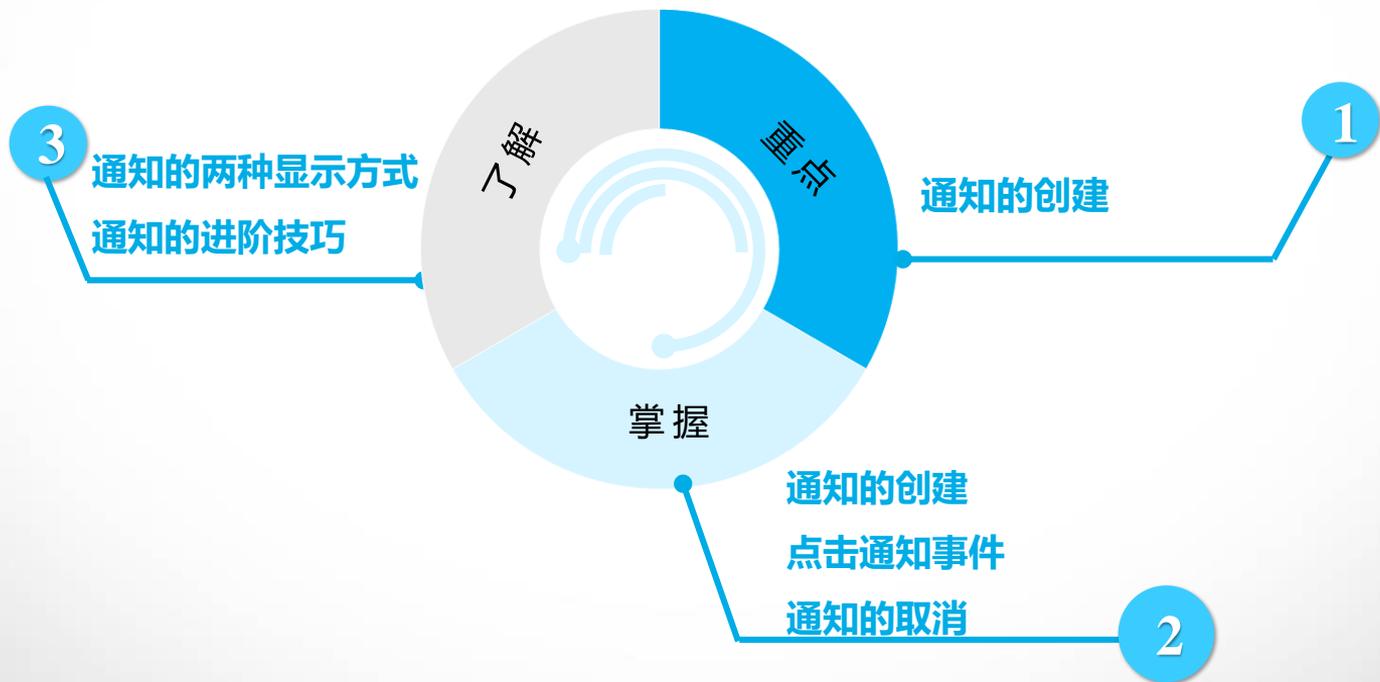


电子信息工程学院

移动终端开发技术

# 丰富你的程序——使用通知

讲 师：陈媛媛





## 通知

点击按钮send notice便发送通知，被当前应用接收到，在状态栏中显示图标，点击后出现详细页面。

# 目录

content

	03	13	18	20	25	Page
通知的两种显示方式						
1						
通知的创建						
2						
通知的点击效果						
3						
取消通知						
4						
进阶技巧						
5						
总结与拓展作业						
6						

1

两种显示方式



# 通知的两种显示方式



## (1) 普通视图

这种风格是notification drawer的标准显示方式。

## (2) 宽视图

指你的notification被展开的时候会显示更大的视图，这种风格是android4.1之后才有的新特性。

下面看一下普通视图的组成

# 通知的两种显示方式



普通视图



1

两种显示方式

2

通知的创建

# 通知的创建

---

(1) 创建NotificationManager对通知进行管理。

```
NotificationManager
```

```
manager=(NotificationManager)getService(NOTIFICATION_SERVICE);
```

# 通知的创建

(2) 使用Builder构造器创建Notification对象。

```
Notification notification=new NotificationCompat.Builder(context).build();
```

进行一些基本的设置:

```
Notification notification=new NotificationCompat.Builder(this)
```

```
    .setContentTitle("this is content title");//用于指定通知的标题内容
```

```
    .setContentText("this is content text");//用于指定通知的正文内容
```

```
    .setWhen(System.currentTimeMillis());//用于指定通知被创建的时间，以毫秒为  
单位
```

```
    .setSmallIcon(R.mipmap.ic_launcher)//用于设置通知的小图标
```

```
    .setLargeIcon(BitmapFactory.decodeResource(getResources(),R.mipmap.ic_lau  
ncher))
```

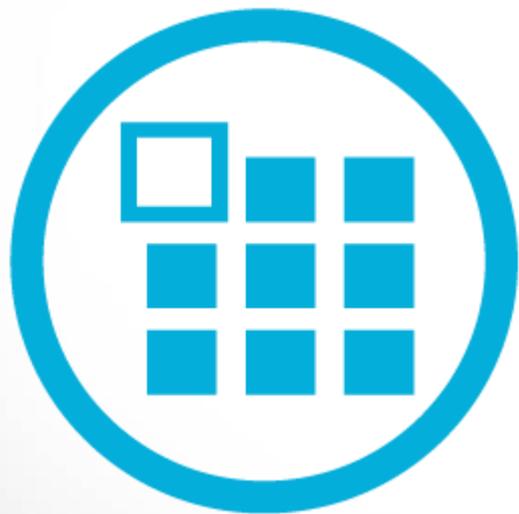
```
    .build();
```

# 通知的创建

---

(3) 调用NotificationManager的notify()方法让通知显示出来。

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



1

两种显示方式

2

通知的创建

3

通知的点击效果

# 通知的点击效果

使用PendingIntent

PendingIntent和Intent有什么区别？



Intent表示立刻执行某个操作，PendingIntent更加倾向于在某个合适的时机去执行某个动作。

# 通知的点击效果

如何得到一个pendingIntent对象？

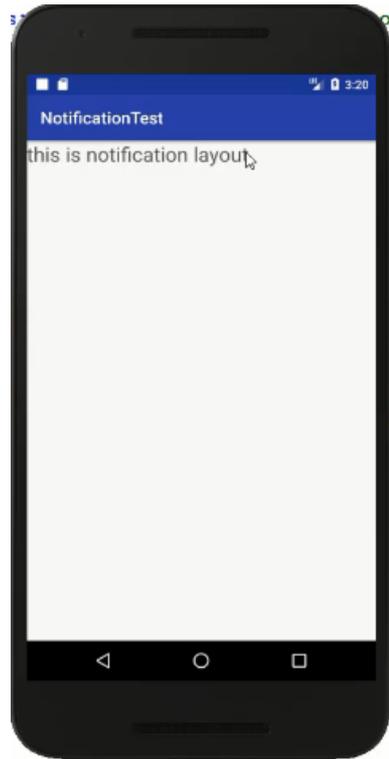
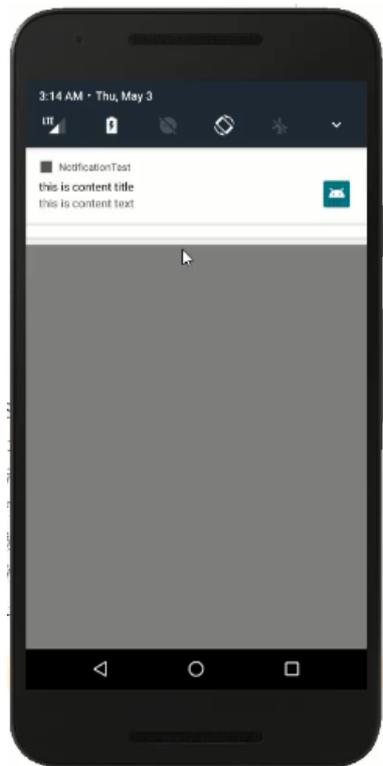
得到一个pendingIntent对象，使用方法类的静态方法 `getActivity(Context, int, Intent, int)`, `getBroadcast(Context, int, Intent, int)`, `getService(Context, int, Intent, int)` 分别对应着Intent的3个行为，跳转到一个activity组件、打开一个广播组件和打开一个服务组件。参数有4个，比较重要的事第三个和第一个，其次是第四个和第二个。可以看到，要得到这个对象，必须传入一个Intent作为参数，必须有context作为参数。其余两个int传入0即可。

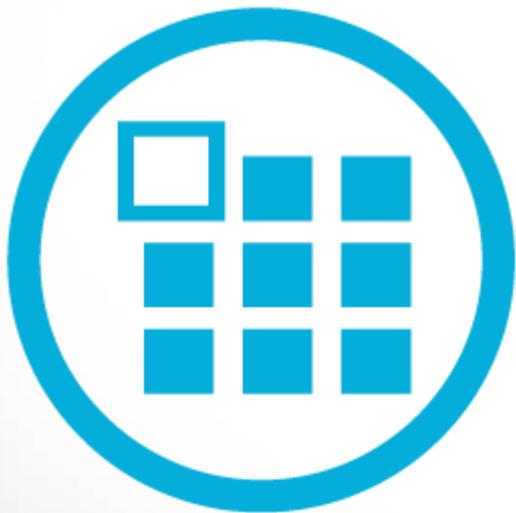
案例：通知的点击效果



# 通知的点击效果

案例





1

两种显示方式

2

通知的创建

3

通知的点击效果

4

通知的取消

# 取消通知

第一种方法：NotificationCompat.Builder中再连一个setAutoCancel()方法。

```
Notification notification=new NotificationCompat.Builder(this)
```

```
...
```

```
.setAutoCancel(true)
```

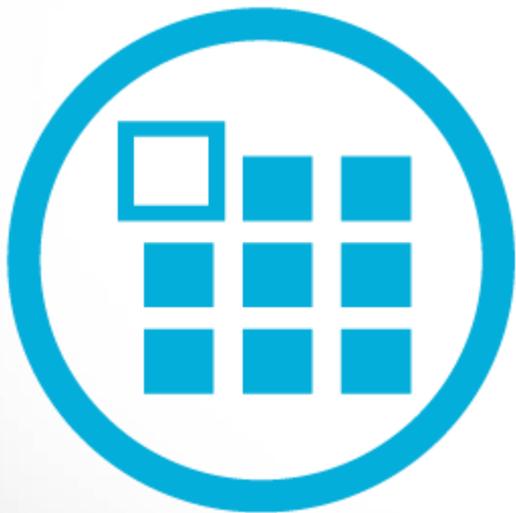
第二种方法：调用cancel()方法

# 取消通知

第二种方法：调用NotificationManager的cancel()方法

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_notification);  
    NotificationManager  
    manager=(NotificationManager)getSystemService(NOTIFICATION_SERVICE);  
    //参数1，这条通知设置的id为1  
    manager.cancel(1);  
}
```

第二种方法



1

两种显示方式

2

通知的创建

3

通知的点击效果

4

通知的取消

5

通知的进阶技巧

# 通知的进阶技巧



NotificationCompat.Builder中提供了非常丰富的API来让我们创建出更加多样的通知效果。如：声音提示、亮光提示灯。

如何实现这些技巧呢

# 通知的进阶技巧

在通知发出的时候播放一段音频：

```
.setSound(Uri.fromFile(new File("/system/media/audio/ringtones/Luna.ogg")))
```

在通知来的时候让手机进行振动：

```
.setVibrate(new long[] {0, 1000, 1000, 1000})
```

不要忘了声明权限

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

在通知来的时候让手机LED灯显示：

```
.setLights(Color.GREEN, 1000, 1000)
```

也可以直接使用通知的默认效果，它会根据当前的手机环境来决定播放什么铃声，以及如何振动，写法如下：

```
.setDefaults(NotificationCompat.DEFAULT_ALL)
```



1

创建通知

---

2

通知的点击效果

---

3

通知的取消及更多技巧

---