



西南交通大学
Southwest Jiaotong University

移动商务应用开发

第8章 广播 Broadcast



8.1 广播概述

8.2 无序广播和有序广播

主要内容

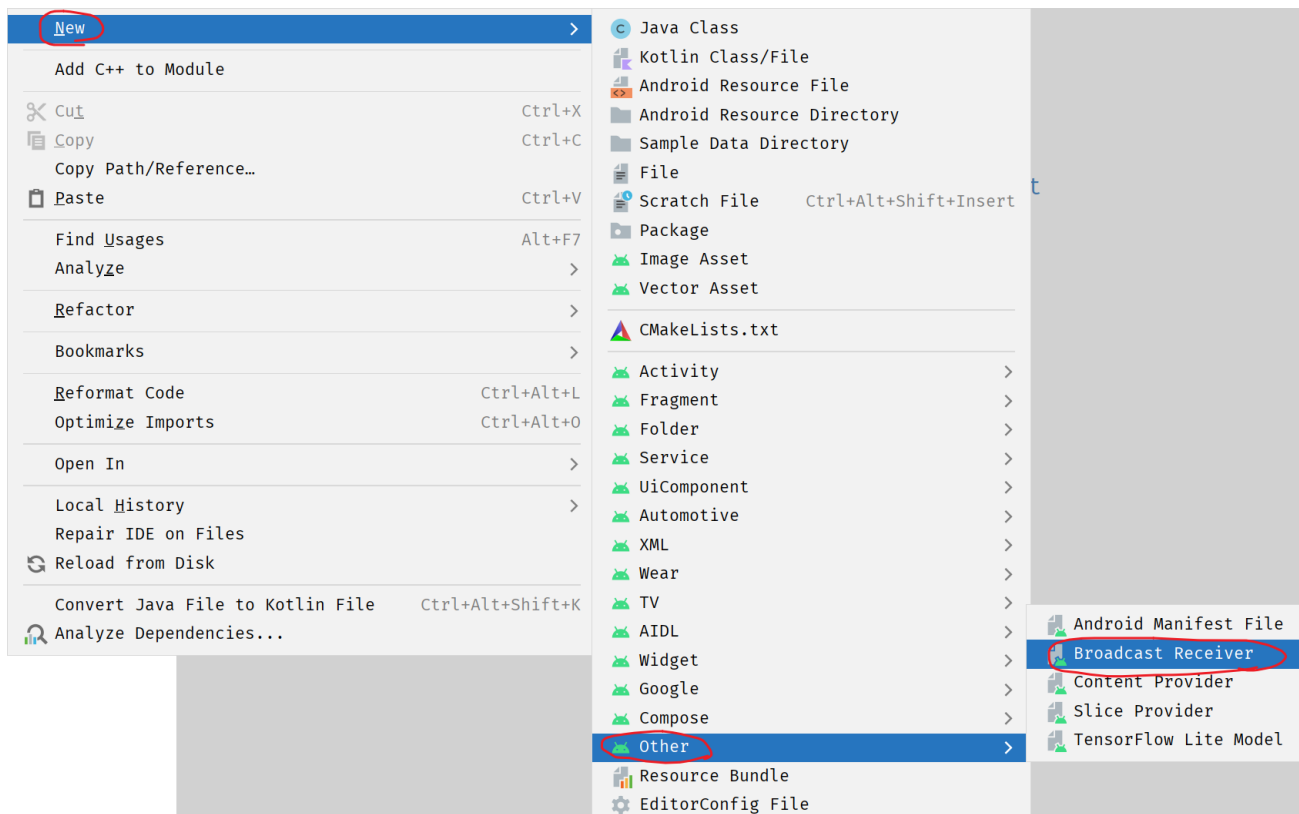
- 广播 (Broadcast) 是Android**四大组件**之一。
- Android应用和应用之间、应用和系统之间可以相互收发广播。
- Android中的广播机制使用了观察者模式，该模式是基于消息的发布/订阅事件模型，该模型中的消息发布者是广播机制中的广播发送者，消息订阅者是广播机制中的广播接收者。



广播的接收



- 接收广播需要用到**广播接收者**（BroadcastReceiver）。
- 广播接收者会对发送出来的广播进行**过滤接收并响应**。





广播的接收



- Android Studio创建BroadcastReceiver的过程中完成了：
- 创建了一个继承BroadcastReceiver的java类
 - 将该BroadcastReceiver在AndroidManifest.xml注册

```
<receiver  
    android:name=".MyReceiver"  
    android:enabled="true"  
    android:exported="true" />
```



广播的接收



➤ 如何使用广播接收者接收广播？

1. 创建自定义的BroadcastReceiver，并重写onReceive()

```
public class MyReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Log.i("myTAG", "我是MyReceiver, 我接受到了广播");  
    }  
}
```

做出响应

获取信息

广播的接收



➤ 如何使用广播接收者接收广播？

1. 创建自定义的BroadcastReceiver，并重写onReceive()
2. 注册广播接收者

```
MyReceiver myReceiver = new MyReceiver();  
IntentFilter intentFilter = new IntentFilter();  
intentFilter.addAction("com.example.broadcastsender.MY_BROADCAST");  
registerReceiver(myReceiver, intentFilter);
```

由Context提供的方法

根据IntentFilter和Permission
来进行广播的过滤接收



广播的接收



➤ 如何使用广播接收者接收广播？

1. 创建自定义的BroadcastReceiver，并重写onReceive()
2. 注册广播接收者
3. 根据需要取消注册

```
protected void onDestroy() {  
    super.onDestroy();  
    unregisterReceiver(myReceiver);  
}
```




广播的发送

- 发送广播需要用到**Intent**。
- 如何利用Intent来发送广播？

```
Intent intent = new Intent();  
intent.setAction("com.example.broadcastsender.MY_BROADCAST");  
sendBroadcast(intent);
```

发送广播

创建Intent对象

- 如何进行广播的匹配？
 - 当发送广播后，系统检查已注册的**BroadcastReceiver**的**IntentFilter**与发送的**Intent**是否匹配。如果匹配，该BroadcastReceiver才能接收到该广播后，并调用该BroadcastReceiver的onReceive()方法。

系统广播和自定义广播



- Android系统内置了很多广播，如系统启动、电量变化、网络变化、飞行模式变化时都会都会发送广播。
 - **系统广播的intent action已经是预定义的。**
- 当系统提供的广播不能满足实际需求时，可以自定义广播
 - **自定义广播的intent action是自定义的。**



➤ 练习1：接收飞行模式变化时系统发送的广播

- intent action 为 "android.intent.action.AIRPLANE_MODE"

➤ 练习2：发送和接收自定义广播

- 应用A发送广播、应用B接收广播
- 自定义 intent action



8.1 广播概述

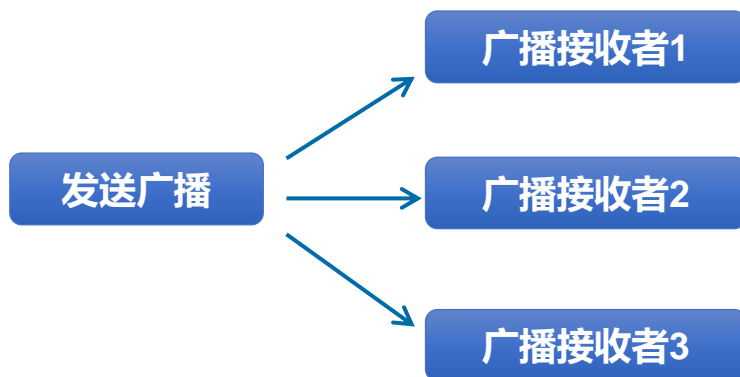
8.2 无序广播和有序广播

主要内容



无序广播和有序广播

- 一个广播可以有**多个**对应的广播接收者接收并进行处理。
- Android系统提供了两种广播类型：**有序广播**和**无序广播**。
- 无序广播（普通广播）
 - 发送无序广播时，所有监听这个广播的广播接收者都会接收到此消息，但**接收的顺序不确定**。

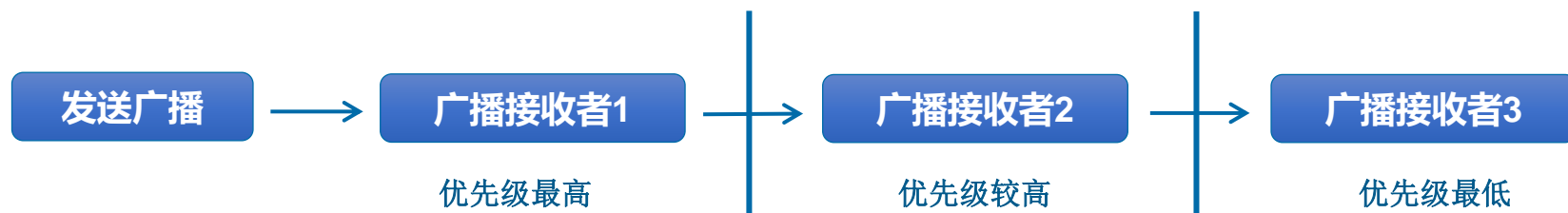


无序广播和有序广播



➤ 有序广播

- 发送有序广播时，同一时刻只有一个广播接收者能接收消息，当一个广播接收者中的逻辑执行完毕后，广播才会继续传递到下一个广播接收者。
- 广播接收者是**有先后顺序**的，优先级高的广播接收者可以先收到广播，并且还可以**截断**正在传递的广播，这样后面的广播接收者就无法收到广播了。



无序广播



➤ 注册广播接收者

```
IntentFilter intentFilter = new IntentFilter();  
intentFilter.addAction("com.example.broadcastsender.MY_BROADCAST");  
MyReceiver1 myReceiver1 = new MyReceiver1();  
registerReceiver(myReceiver1, intentFilter);
```

➤ 发送广播

```
Intent intent = new Intent();  
intent.setAction("com.example.broadcastsender.MY_BROADCAST");  
sendBroadcast(intent);
```

↓
发送无序广播

有序广播



➤ 注册广播接收者

```
MyReceiver1 myReceiver1 = new MyReceiver1();  
IntentFilter intentFilter1 = new IntentFilter();  
intentFilter1.addAction("com.example.broadcastsender.MY_BROADCAST");  
intentFilter1.setPriority(10);  
registerReceiver(myReceiver1, intentFilter1);
```

→ 数值越大，优先级越高

➤ 发送广播

```
Intent intent = new Intent();  
intent.setAction("com.example.broadcastsender.MY_BROADCAST");  
sendOrderedBroadcast(intent, null);
```

↓
发送有序广播

有序广播



➤ 截断广播

```
@Override
public void onReceive(Context context, Intent intent) {
    Log.i("myTAG", "我是MyReceiver2, 我接受到了广播");
    abortBroadcast(); // 截断广播
}
```

练习



- 练习1：发送和接收无序广播
- 练习2：发送和接收有序广播



8.1 广播概述

8.2 无序广播和有序广播

主要内容

本章作业

- 发送和接收有序广播
 - 截图输出日志

作业提交方式

- 截图，文件名：学号+姓名+第8章作业
- 邮件给助教，主题：学号+姓名+第8章作业



➤ 作业示例

13:28:01.071 myTAG

13:28:11.530 myTAG

13:28:11.531 myTAG

13:28:11.532 myTAG

I 学号+姓名:

注册了3个广播接收者，权重分别为10、20、30

I 我是MyReceiver3，我接收到了广播

I 我是MyReceiver2，我接收到了广播

I 我是MyReceiver1，我接收到了广播