



河北师范大学软件学院
Software College of Hebei Normal University



Android 基础开发

第十一章 BroadcastReceiver



Java与移动智能设备开发



教学目标

- 掌握Android中BroadcastReceiver的使用



目录

1 BroadcastReceiver简介

2 接收广播

3 发送广播



生活中的广播





广播机制的总结

- 广播发送者
 - 可发送一种或多种广播
 - 不关心是谁接收
 - 具有实时性
- 广播接收者
 - 可接收一种或多种广播
 - 不关心是谁发送
 - 具有实时性

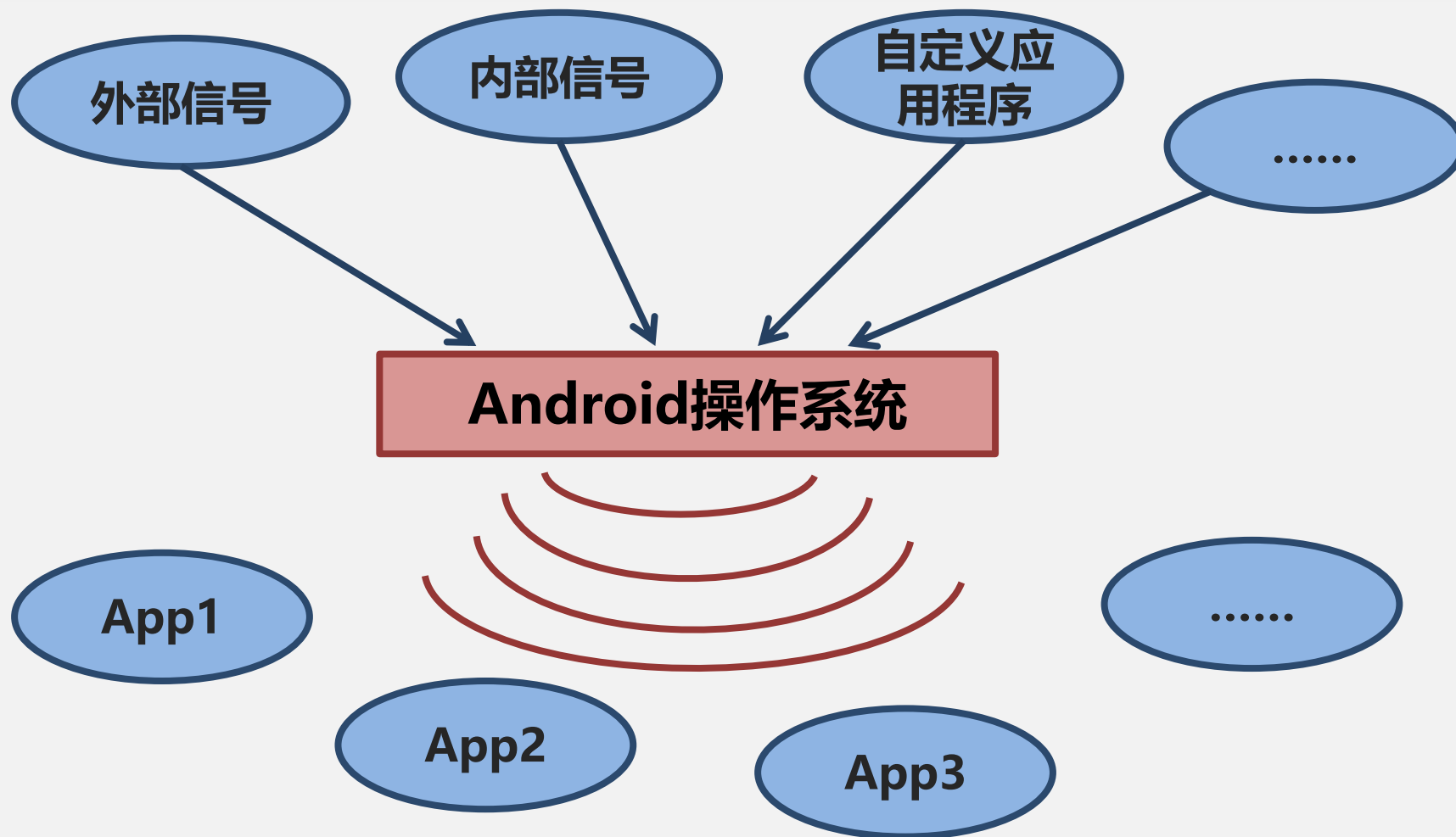


BroadcastReceiver简介

- BroadcastReceiver组件本质上是一种**全局的监听器**，因此它的主要作用是实现系统间不同组件之间的通讯。
- 例如：
 - 通过startService启动的服务如何返回状态给Activity。



Android广播机制原理图





Android中广播机制的介绍

- Android手机中有很多应用采用广播机制：
 - 电话的接听和拨打
 - 短信的接收和发送
 - 电池的状态
 - 系统的闹钟
 - 手机连接电脑
 - 手机脱离电脑
 -



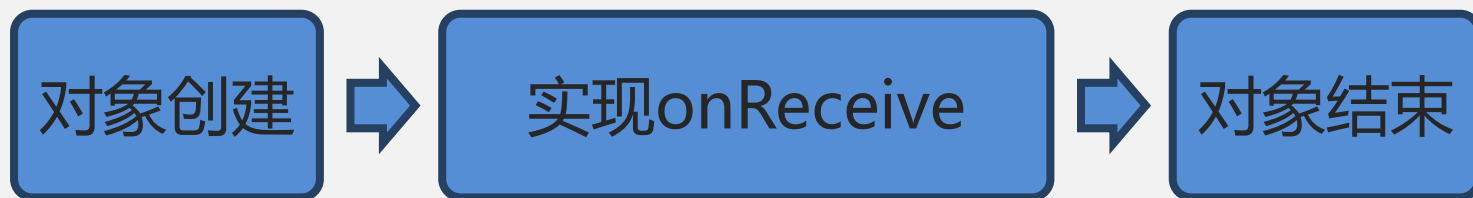
目录

- 1 BroadcastReceiver简介
- 2 接收广播
- 3 发送广播



广播接收者

- BroadcastReceiver是Android中接收广播的类。
- 在BroadcastReceiver中可以接收广播，并做出相应处理，不会**直接显示页面**，一般处理为：
 - 启动Activity来显示相应内容。
 - 使用NotificationManager来通知用户。
 - 广播给其他应用程序。
- BroadcastReceiver的生命周期：





广播接收者

- 使用步骤：
 - 自定义一个类继承BroadcastReceiver，复写onReceive()方法。
 - 设置BroadcastReceiver能够接收的广播类型并注册该BroadcastReceiver（静态注册/动态注册）。
 - [添加相应权限]。
 - 调用Context的sendBroadcast()或sendOrderedBroadcast()启动BroadcastReceiver。



BroadcastReceiver

```
// 自定义BroadcastReceiver
public class MyReceiver extends BroadcastReceiver {
    public void onReceive(Context context, Intent intent) {
        Toast.makeText(context, "接收到的Intent的Action为 :
" + intent.getAction() + "\n消息的内容是：" +
intent.getStringExtra("msg"), Toast.LENGTH_SHORT)
            .show();
    }
}
```



BroadcastReceiver

```
// 设置BroadcastReceiver能够接收的广播类型
IntentFilter filter = new
IntentFilter("net.onest.lww.CUSTOM_BROADCAST");
// 注册BroadcastReceiver (动态注册方式)
MyReceiver myReceiver = new MyReceiver();
registerReceiver(myReceiver, filter);
```

```
// 发送广播
Intent intent = new Intent(
    "net.onest.lww.CUSTOM_BROADCAST");
intent.putExtra("msg", "测试携带内容");
sendBroadcast(intent);
```



广播接收者

- 注意：
 - **onReceive方法中不能加入比较耗时的操作**，否则系统会认为程序无响应，如果一定要执行耗时的操作的话，一般通过Intent启动一个Service来完成。



创建一个BroadcastReceiver

- 注册BroadcastReceiver的两种方式：

- 静态注册

- 动态注册

```
//在AndroidManifest.xml文件当中进行注册
<receiver android:name=".MyAndroidReceiver">
    <intent-filter>
        <action android:name="广播类型" />
    </intent-filter>
</receiver>
```

//在代码中注册

```
IntentFilter filter = new IntentFilter("广播类型");
MyAndroidReceiver myAndroidReceiver = new MyAndroidReceiver();
registerReceiver(myAndroidReceiver , filter);
```

//在代码中注销

```
unregisterReceiver(MyAndroidReceiver);
```



创建一个BroadcastReceiver



- 注册BroadcastReceiver的两种方式区别：
 - 静态注册
 - 静态注册的广播，即使Activity销毁了，仍然可以收到广播。
 - 动态注册
 - 动态注册的广播会受Activity的生命周期的影响，当Activity销毁的时候，广播就失效了。



创建一个BroadcastReceiver

- 注册BroadcastReceiver
 - 静态注册：常驻型，也就是当应用程序关闭后，仍然会接收广播消息，程序也会被系统调用运行。一旦安装应用就注册，不可注销。(Android API26 取消大部分自定义的静态广播注册)
 - 动态注册：不是常驻型，可以手工注册，手工注销。
- 在BroadcastReceiver的onReceive()中可使用abortBroadcast()中止广播，这样其他接收同样广播的应用就不会再接收到这个广播了。



常见广播类型

Constants		
String	<code>ACTION_AIRPLANE_MODE_CHANGED</code>	Broadcast Action: The user has switched the phone into or out of Airplane
String	<code>ACTION_ALL_APPS</code>	Activity Action: List all available applications Input: Nothing.
String	<code>ACTION_ANSWER</code>	Activity Action: Handle an incoming phone call.
String	<code>ACTION_APP_ERROR</code>	Activity Action: The user pressed the "Report" button in the crash/ANR dia
String	<code>ACTION_ATTACH_DATA</code>	Used to indicate that some piece of data should be attached to some other
String	<code>ACTION_BATTERY_CHANGED</code>	Broadcast Action: This is a <i>sticky broadcast</i> containing the charging state, information about the battery.
String	<code>ACTION_BATTERY_LOW</code>	Broadcast Action: Indicates low battery condition on the device.
String	<code>ACTION_BATTERY_OKAY</code>	Broadcast Action: Indicates the battery is now okay after being low.
String	<code>ACTION_BOOT_COMPLETED</code>	Broadcast Action: This is broadcast once, after the system has finished bo
String	<code>ACTION_BUG_REPORT</code>	Activity Action: Show activity for reporting a bug.
String	<code>ACTION_CALL</code>	Activity Action: Perform a call to someone specified by the data.
String	<code>ACTION_CALL_BUTTON</code>	Activity Action: The user pressed the "call" button to go to the dialer or oth placing a call.
String	<code>ACTION_CAMERA_BUTTON</code>	Broadcast Action: The "Camera Button" was pressed.
String	<code>ACTION_CHOOSER</code>	Activity Action: Display an activity chooser, allowing the user to pick what t proceeding.

http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW



Android中的权限

- Android是一个以Linux为核心的操作系统，所以也像Linux一样拥有严格的权限管理机制。
- 一般的程序在其AndroidManifest文件中要声明其应用程序的权限，例如：打电话需要有打电话的权限，发短信要有发短信的权限。

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



常见的Android权限

- android.permission.ACCESS_FINE_LOCATION允许一个程序访问精良位置(如GPS)
- android.permission.ACCESS_WIFI_STATE允许程序访问Wi-Fi网络状态信息
- android.permission.BATTERY_STATS允许程序更新手机电池统计信息
- android.permission.BLUETOOTH允许程序连接到已配对的蓝牙设备
- android.permission.CAMERA请求访问使用照相设备
- android.permission.CHANGE_WIFI_STATE允许程序改变Wi-Fi连接状态
- android.permission.DEVICE_POWER允许访问底层电源管理
- android.permission.INTERNET允许程序打开网络套接字
- android.permission.MOUNT_UNMOUNT_FILESYSTEMS允许挂载和反挂载文件系统可移动存储
- android.permission.READ_CONTACTS允许程序读取用户联系人数据
- android.permission.READ_LOGS允许程序读取底层系统日志文件
- android.permission.REBOOT请求能够重新启动设备



接收系统广播



- 接收电话广播demo
- 接收短信广播demo
- 电话黑名单



目录

- 1 BroadcastReceiver简介
- 2 接收广播
- 3 发送广播



发送广播

普通广播

完全异步

无法终止

无法传递处理结果

`sendBroadcast`

有序广播

顺序接收
`priority`属性

可以终止
`abortBroadcast`

可以传递处理结果
`setResultExtras`
`getResultExtras`

`sendOrderedBroadcast`



发送广播

- 广播的类型：
 - 普通广播：完全异步，可在同一时刻被所有接受者接收到，接受者无法将处理结果传递给下一个接受者，无法终止Broadcast的传播
 - 有序广播：接收者按照优先级依次接收广播，优先级声明在intent-filter元素的priority属性中。广播可被终止，可以将处理结果传给下一个接收者。
 - 数越大优先级别越高，最大值是2147483647
 - 可以调用IntentFilter对象的setPriority()进行设置
- PS:发送广播一般是自定义广播，一般系统会限制你发送系统广播。



发送广播



- 广播的优先级设置

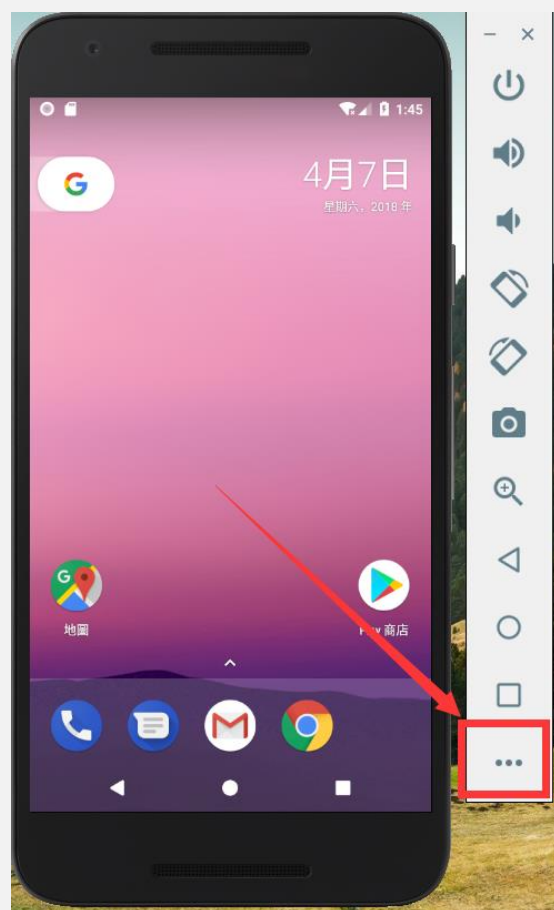
//在AndroidManifest.xml文件当中进行注册

```
<receiver android:name="包名.广播接收者类名" >  
    <intent-filter android:priority="广播拦截的优先级(最大:2147483647)" >  
        <action android:name="广播监听的动作 可以是自定义的或者系统广播" />  
    </intent-filter>  
</receiver>
```



模拟器控制（补充）

- 使用模拟器模拟接收短信与电话处理



Extended controls - Nexus_5X_API_26:5554

- Location
- Cellular
- Battery
- Phone**
- Directional pad
- Microphone
- Fingerprint
- Virtual sensors
- Bug report
- Record screen
- Google Play
- Settings
- Help

From (650) 555-1212

HOLD CALL **CALL DEVICE**

SMS message

Oreo's a slam dunk!

SEND MESSAGE

向模拟器打电话

短信内容

向模拟器发送短信



内容回顾

- 1 BroadcastReceiver简介
- 2 接受系统广播
- 3 发送广播



Thank you!

