

BroadcastReceiver (广播)

1,register for system or application events

2,ACTION_BOOT_COMPLETED system event 监听是否开机

3, 典型的应用: 接受Android自带的短信, 电话等广播, 进行电话, 短信拦截等。

4, 通知, 通信, 数据传递。

5, 注册方式: 动态注册, xml中注册。区别为作用范围不同:

程序动态注册的接收者只在程序运行过程中有效, 而在xml注册的接收者不管你的程序有没有启动有会起作用

There are following two important steps to make BroadcastReceiver works for the system broadcasted intents –

- Creating the Broadcast Receiver.
- Registering Broadcast Receiver

There are several system generated events defined as final static fields in the **Intent** class. The following table lists a few important system events.

Event Constant	Description
android.intent.action.BATTERY_CHANGED	充电状态, 电量, 其他关于电池的信息
android.intent.action.BATTERY_LOW	电量不足
android.intent.action.BATTERY_OKAY	之前电量不足现在良好
android.intent.action.BOOT_COMPLETED	系统启动完成 (开机检测)
android.intent.action.BUG_REPORT	在activity中显示一个错误
android.intent.action.CALL	拨打电话
android.intent.action.CALL_BUTTON	拨打电话页面
android.intent.action.DATE_CHANGED	数据改变
android.intent.action.REBOOT	设备重新启动

There are several system generated events defined as final static fields in the **Intent** class. The following table lists a few important system events.

Event Constant	Description
android.intent.action.BATTERY_CHANGED	Sticky broadcast containing the charging state, level, and other information about the battery.
android.intent.action.BATTERY_LOW	Indicates low battery condition on the device.
android.intent.action.BATTERY_OKAY	Indicates the battery is now okay after being low.
android.intent.action.BOOT_COMPLETED	This is broadcast once, after the system has finished booting.
android.intent.action.BUG_REPORT	Show activity for reporting a bug.
android.intent.action.CALL	Perform a call to someone specified by the data.
android.intent.action.CALL_BUTTON	The user pressed the "call" button to go to the dialer or other appropriate UI for placing a call.
android.intent.action.DATE_CHANGED	The date has changed.
android.intent.action.REBOOT	Have the device reboot.

sendBroadcast(): 只接受一次, 后面即使再处于那个状态, 也不会接收到广播。

sendStickyBroadcast(Intent): 可以接受多次, 即每次处于哪个状态, 都会接受广播。

动态使用广播:

- 1, 创建类继承自BroadcastReceiver.*
- 2, 创建一个方法, 设置setaction属性 (通过intent) 指定标示.*
- 3, 在注册文件中注册使用.*

生命周期:

生命周期只有十秒左右, 如果在onReceiver中做超过十秒的事情会报错。

广播分类:

同步广播: (normal broadcasts) 发送一个广播, 所有监听该广播的广播接收者都可以监听到该广播。

异步广播: 当处理完之后的Intent, 依然存在, 这时候registerReceiver(BroadcastReceiver,IntentFilter) 还能收到他的值, 直到你把它去掉,不能将处理结果传给下一个接收者,无法终止广播.

有序广播: (ordered broadcasts)按照接收者的优先级顺序接收广播,优先级在intent-filter中的priority 中声明,-1000到1000之间,值越大,优先级越高.可以终止广播意图的继续传播.接收者可以篡改内容.

使用sendBroadcast()或sendStickyBroadcast()方法发出去的Intent, 所有满足条件的BroadcastReceiver都会随机地执行其onReceive()方法

普通广播的发送和接收:

```
sendBroadcast(intent);
```

```
Intent intent = new Intent("cn.lenovo.yangguang");
sendBroadcast(intent);
```

priority: 这个是**AndroidManifest.xml**中intent-filter的参数。

```
<receiver android:name= ".MyBroadcastReceiver">
<intent-filter android:priority= "1000">
```

```
<action android:name="cn.lenovo.yangguangfu"/>
</intent-filter>
</receiver>
```

sendOrderedBroadcast(intent, receiverPermission);

1, 他决定该广播的级别, 级别数值是在-1000到1000之间, 值越大, 优先级越高;

2, 同级别接收是先后是随机的; 级别低的收到广播;

3, 在Android系统中只要监听该广播的接收者, 都能够收到sendBroadcast(intent)发出的广播

3, 不能截断广播的继续传播,

4, 实验现象, 在这个方法发来的广播中, 代码注册方式中, 收到的广播的先后和注明优先级最高的他们的先后是随机。如果都没有优先级, 代码注册收到为最先。

异步广播的发送和接收:

sendStickyBroadcast(intent);

当处理完之后的Intent, 依然存在, 直到你把它去掉。

发这个广播需要权限<uses-permission android:name="android.permission.BROADCAST_STICKY"/>

去掉是用这个方法removeStickyBroadcast(intent);但别忘了在执行这个方法的应用里面AndroidManifest.xml同样要加上面的权限;

sendStickyOrderedBroadcast(intent, resultReceiver, scheduler,
initialCode, initialData, initialExtras)

这个方法具有有序广播的特性也有异步广播的特性;

发送这个广播要: <uses-permission android:name="android.permission.BROADCAST_STICKY"/>这个权限。才能使用这个方法。如果您并不拥有该权限, 将抛出SecurityException的。

实验现象 (sendStickyOrderedBroadcast () 中), 在这个方法发来的广播中, 代码注册方式中, 收到广播先后次序为: 注明优先级的、代码注册的、没有优先级的; 如果都没有优先级, 代码注册收到为最先

广播注册与注销

代码中注册广播:

注册广播方法一: registerReceiver(BroadcastReceiver receiver, IntentFilter filter), 第一个参数是我们要处理广播的BroadcastReceiver (广播接收者, 可以是系统的, 也可以是自定义的); 第二个参数是意图过滤器。

注册广播方法二: registerReceiver(receiver, filter, broadcastPermission, scheduler), 第一个参数是BroadcastReceiver (广播接收者, 可以是系统的, 也可以是自定义的); 第二个参数是意图过滤器; 第三个参数是广播权限; 第四个参数是Handler;

注意: 权限重复现象, 如果功能清单文件里注册了权限, 在该方法再注册, 则receiver无法收到广播, 如果功能清单文件里没有注册了权限, 该方法注册也无法收到。当该方法没有注册权限, 功能清单里注册的时候, receiver能收到广播。

总结: 在Activity中代码注册广播建议在: onResume()中注册;