

Broadcast Receiver on Android

April 04, 2016

前言

Broadcast Receiver是Android组件中最基本也是最为常见用的四大组件之一。

Broadcast Receiver, [广播接收者](#), 顾名思义, 是用来接收来自系统和应用中的广播。这种组件本质上是一种全局的监听器, 用于监听系统全局的广播消息。

目录

1. Broadcast Receiver的生命周期
2. Broadcast的类型
3. Broadcast Receiver的注册方式
4. Broadcast Receiver的使用步骤

一、Broadcast Receiver的生命周期

BroadcastReceiver 有一个回调方法：void onReceive(Context curContext, Intent broadcastMsg)。当一个广播消息到达接收者时，Android 调用它的 onReceive() 方法并传递给它包含消息的 Intent 对象。BroadcastReceiver 被认为仅当它执行这个方法时是活跃的。当 onReceive() 返回后，它是不活跃的。

Android系统的很多消息(如系统启动,新短信,来电话等)都通过 BroadcastReceiver来分发。BroadcastReceiver的生命周期是短暂的，而且是消息一到达则创建执行完毕就立刻销毁的。

拥有一个活跃状态的广播接收器的进程被保护起来而不会被杀死，但仅拥有失活状态组件的进程则会在其它进程需要它所占有的内存的时候随时被杀掉。所以，如果响应一个广播信息需要很长的一段时间，我们一般会将其纳入一个衍生的线程中去完成，而不是在主线程内完成它，从而保证用户交互过程的流畅。

这带来一个问题，当一个广播消息的响应时费时的，因此应该在独立的线程中做这些事，远离用户界面或者其它组件运行的主线程。如果 onReceive() 衍生线程然后返回，整个进程，包括新的线程，被判定为不活跃的（除非进程中的其它应用程序组件是活跃的），否则将使它处于被杀的危机。解决这个问题的方法是 onReceive() 启动一个 Service，让 Service 做这个工作，因此系统知道进程中有活跃的工作在做。

通常某个应用或系统本身在某些事件（电池电量不足、来电来短信）来临时会广播一个 Intent 出去，通过注册一个 BroadcastReceiver 来监听到这些 Intent，并获取其中广播的数据。

二、Broadcast的类型

Broadcast的类型有两种：普通广播和有序广播。

Normal broadcasts（普通广播）：Normal broadcasts是完全异步的可以同一时间被所有的接收者接收到。通常每个接收者都无需等待即可以接收到广播，接收者相互之间不会有影响。消息的传递效率比较高。但缺点是接收者不能讲接收的消息的处理信息传递给下一个接收者也不能停止消息的传播。

Ordered broadcasts（有序广播）：Ordered broadcasts的接收者按照一定的优先级进行消息的接收。如：A,B,C的优先级依次降低，那么消息先传递给A，在传递给B，最后传递给C。优先级别声明在中，取值为[-1000,1000]数值越大优先级别越高。优先级也可通过filter.setPriority(10)方式设置。另外Ordered broadcasts的接收者可以通过abortBroadcast()的方式取消广播的传播，也可以通过setResultData和setResultExtras方法将处理的结果存入到Broadcast中，传递给下一个接收者。然后，下一个接收者通过getResultData()和getResultExtras(true)接收高优先级的接收者存入的数据。

三、Broadcast Receiver的注册方式

注册有两种方式：

1、静态注册：这种方法是在配置AndroidManifest.xml配置文件中的application里面定义receiver并设置要接收的action。通过这种方式注册的广播为常驻型广播，也就是说如果应用程序关闭了，有相应事件触发，程序还是会被系统自动调用运行。如：

```
<!-- 在配置文件中注册BroadcastReceiver能够匹配的Intent -->
<receiver android:name="com.example.test.MyBroadcastReceiver">
    <intent-filter>
        </action>
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>
```

2、动态注册：这种方法是通过代码在Java文件中进行注册。通过这种方式注册的广播为非常驻型广播，即它会跟随Activity的生命周期，所以在Activity结束前我们需要调用unregisterReceiver(receiver)方法移除它，否则会报异常。

```
//通过代码的方式动态注册MyBroadcastReceiver
MyBroadcastReceiver receiver=new MyBroadcastReceiver();
```

```
IntentFilter filter=new IntentFilter();  
filter.addAction("android.intent.action.MyBroadcastReceiver");  
//注册receiver  
registerReceiver(receiver, filter);
```

要接收某些action，需要在AndroidManifest.xml里面添加相应的permission。

四、Broadcast Receiver的使用步骤

1、创建BroadcastReceiver的子类

由于BroadcastReceiver本质上是一种监听器，所以创建BroadcastReceiver的方法也非常简单，只需要创建一个BroadcastReceiver的子类然后重写onReceive (Context context, Intent intent)方法即可。

2、注册BroadcastReceiver

一旦实现了BroadcastReceiver，接下就应该指定该BroadcastReceiver能匹配的Intent即注册BroadcastReceiver。

另外，这是发送广播的代码：

```
Intent intent = new Intent(String action);  
intent.setAction(String action);  
sendBroadcast(Intent);
```
