

# Android移动开发课程

## 实验指导书- BroadcastReceiver部分

### 【实验目的】

初步了解Android 的 Notification 组件

### 【实验设计】

本次实验包括两个验证实验和一个自选实验。

其中验证实验为个人实验，已经提供源代码、操作步骤、实验指导视频

由于NetWork涉及网络通讯，因此本实验提供了一个Jsp编写的服务器，源代码已经给出，可以在Tomcat下运行。

要求：

按照实验步骤完成，

以个人为单位提交，提交实验报告一份，实验报告需要回答指导书中问题。

自选实验为小组实验，建议2-4人组队，提供了参考选题和参考资料。

要求：

以小组为单位提交，提交实验报告一份、源代码一份、可以执行的APK文件一个。

### 【实验内容】

#### 验证实验一：了解广播接收和接收器的静态注册

##### 1.了解广播接收。

自定义广播接收器继承于基类BroadcastReceiver，并且实现抽象方法

onReceiver(content intent)，当广播接收器接受到广播的时候，自动回调 onReceiver方法。广播器的生命周期非常短暂，当广播器收到广播的时候开始创建，执行完receiver方法后就销毁。默认情况下，广播接收器工作在主线程（UI线程），因此，广播接收器不能执行耗时操作，否则会引发ANR，参考时间：10秒。

了解自定义广播接收器可以执行的操作。自定义广播接收器可以执行的操作有:Toast、发布通知栏消息、对话框（当context时Activity的时候）、发送广播、开启服务、开启新的Activity

## 2. 掌握广播的静态注册。

静态注册通过AndroidManifest进行，和其他三大组件一样，静态注册的广播在应用安装时由系统自动完成注册。使用intentfilter对接受的广播进行过滤。知道内部静态类的广播注册，在内部类所在类和内部类之间以美元符号分隔。

程序退出以后，静态注册的广播接收器是否能够接受广播并且处理：

1. 在3.1以后，增加了FLAG\_INCLUDE\_STOP\_PACKAGE(包含已经停止的包)和FLAG\_EXCLUDE\_STOPPED\_PACKAGE(不包含已经停止的包)，3.1以后系统对大多数广播自动增加FLAG\_EXCLUDE\_STOPPED\_PACKAGE的FLAG.

2.了解到Android 7.0 移除了三项隐式广播：a) 面向Android 7.0及以上开发的用户不会收到CONNECTIVITY\_ACTION广播，即使在Manifest中进行了注册，注：监听网络状态利用的是CONNECTIVITY\_ACTION广播。在前台运行的应用可以通过动态注册监听CONNECTIVITY\_CHANGE。b) 应用无法发送或者接收ACTION\_NEW\_PICTURE或者ACTION\_NEW\_VIDEO广播.

3. 了解到Android 8.0 中，无法使用清单声明隐式广播的接收者，除了一些免除该限制的广播。(并不是说静态注册无法使用，而是静态注册收不到大多数隐式广播)

3. 创建一个静态注册的接收器：在exp这个package下右击  
"New->Other->BroadcastReceiver",填写文件名,重写receiver方法。使用Manifest修改receiver，增加intentfilter.

4. 在example下提供了本实验源码，分别是Exp1Receiver和Exp1Activity两个文件。

点击"发送一条隐式无序广播"按钮、"发送一条显式无序广播"按钮、"发送一条隐式广播,接收者为静态内部类"按钮、"发送一条显示广播,接收者为静态内部类"按钮，通过Logcat分析执行情况。（过滤条件：Level为Verbose，关键词为Exp1）

5. 修改Exp1Receiver->onCreated中的execTime模拟广播的执行，了解ANR时间。

本次小实验提交内容： 截取Logcat、关于这些Logcat的分析。

## 验证实验二：掌握接收器的动态注册

1. 创建 IntentFilter 并调用 registerReceiver(BroadcastReceiver, IntentFilter) 来注册接

收器。只要注册Content有效，动态注册的接收器就会接收广播。例如，如果您在 Activity 上下文中注册，只要 Activity 没有被销毁，接收器就会收到广播。如果您在ApplicationContext中注册，只要应用在运行，接收器就会收到广播。

2. 在example下提供了本实验源码，分别是Exp2Receiver和Exp2Activity两个文件。

打开实验二，下拉状态栏，将移动数据关闭，观察界面，了解broadcastreceiver中如果context是一个Activity，则可以使用Dialog。

多次点击"注册一个监听exp2.1的广播"按钮和"发送一条exp2.1的广播"按钮，观察Logcat（过滤条件：Level为Info，关键词为Exp2）。

多次点击"注册一个监听exp2.2的广播"按钮和"发送一条exp2.2的广播"按钮，观察Logcat（过滤条件：Level为Info，关键词为Exp2）。

3.点击"发送一条exp2.3的广播"按钮，观察界面和Logcat，了解receiver中可以使用startActivity(不建议)、startService、sendReceiver（过滤条件：Level为Info，关键词为Exp2）

4.点击"发送一条exp2.4的广播"按钮和"发送一条exp2.5的广播"按钮，观察界面和Logcat，了解可以注册为一个receiver对象注册多个监听。（过滤条件：Level为Info，关键词为Exp2）

本次小实验提交内容： 截取Logcat、关于这些Logcat的分析。

### 验证实验三：掌握广播的发送

1. Android 为应用提供三种方式来发送广播：

sendOrderedBroadcast(Intent, String) 方法一次向一个接收器发送广播。当接收器逐个顺序执行时，接收器可以向下传递结果，也可以完全中止广播，使其不再传递给其他接收器。接收器的运行顺序可以通过匹配的 intent-filter 的 android:priority 属性来控制；具有相同优先级的接收器将按随机顺序运行。

sendBroadcast(Intent) 方法会按随机的顺序向所有接收器发送广播。这称为常规广播。这种方法效率更高，但也意味着接收器无法从其他接收器读取结果，无法传递从广播中收到的数据，也无法中止广播。

LocalBroadcastManager.sendBroadcast 方法会将广播发送给与发送器位于同一应用

中的接收器。如果您不需要跨应用发送广播，请使用本地广播。这种实现方法的效率更高（无需进行进程间通信），而且无需担心其他应用在收发广播时带来的任何安全问题。

2. 在example下提供了本实验源码，分别是Exp3Receiver和Exp3Activity两个文件。

点击"发送一条exp3.2的标准广播"按钮，查看Logcat显示的情况（过滤条件：Level为Info，关键词为Exp2）

点击"发送一条exp3.3的有序广播"按钮，查看Logcat显示的情况（过滤条件：Level为Info，关键词为Exp2）

分别取消receiver中的abort前的注释，然后再次点击"发送一条exp3.3的有序广播"按钮，查看Logcat显示的情况。

3. 打开BroadcastReceiver2并在模拟器上运行这个应用，点击"实验三"。

点击BroadcastReceiver1-实验三-中的"发送一条exp3.2的标准广播"按钮，查看BroadcastReceiver2的显示情况。

点击BroadcastReceiver1-实验三-中的"发送一条exp3.2的有序广播"按钮，查看BroadcastReceiver2的显示情况。

了解到广播的发送是系统级别的。

本次小实验提交内容： 截取Logcat、关于这些Logcat的分析。

## 验证实验四：掌握本地广播

1. 采用是sendBroadcast(Intent)、sendOrderedBroadcast(Intent, String)发送的广播均为系统全局广播，发出的广播可以被其他任何应用程序接收到，也可以接收来自其他任何应用程序的广播。这两个方法主要是向系统核心服务AMS发起了一个异步请求，通过AMS发送全局广播。

2. Android还引入了一套本地广播机制，使用LocalBroadcastManager来管理本地广播。

在example下提供了本实验源码，分别是Exp4Receiver和Exp4Activity两个文件。

点击"发送一条本地广播"按钮，查看Logcat显示的情况（过滤条件：Level为Info，关键词为Exp2）

本次小实验提交内容： 截取Logcat、关于这些Logcat的分析。

## 自选实验

从以下选项中选择一个完成，或者自选实验：

### 1. 设计一个具有下线功能的广播：

在全家桶里通常共享一个账户，当一个账户选择从设备中登出的时候，该全家桶里其他正在运行的程序也应该同步登出。利用广播机制，在登出账户时强制下线。

本实验可以组队，建议2-4个人完成，每一个人完成一个应用程序，对于每个应用程序有应该：拥有一个发送全局广播的按钮，其他应用可以接收该广播；拥有一个接收器，动态注册，并能利用context弹出已下线对话框。

加分项：采用了广播权限设计，参考权限

<https://developer.android.com/guide/components/broadcasts#restrict-broadcasts-permissions>