**Android broadcast机制**

参考：<https://blog.csdn.net/Luoshengyang/article/details/6730748>

1. Broadcast机制分为两步：
2. 广播的接收者把广播接收器注册到ActivityManagerService中；
3. 广播的发送者把广播发送到ActivityManagerService中，由ActivityManagerService去查找注册了这个广播的接收者，并把广播分发给它们。
4. 广播接收器的注册过程

参考：<https://blog.csdn.net/luoshengyang/article/details/6737352>

1. 通过ContextWrapper.registerReceiver()并最终调用ContextImpl.registerReceiver()；
2. 在ContextImpl.registerReceiver()里通过mMainThread.getHandler获取到用于发送广播消息的handler；通过ReceiverDispatcher.getIIntentReceiver()获取一个Binder并将其传给ActivityManagerService，用于接收广播；
3. 通过ActivityManagerService.registerReceiver将上一步获取的应用程序Binder记录到ActivityManagerService中。先调用getStickiesLocked()查找sticky intent列表是否存在相同action类型的广播接收器，存在就直接返回；否则将其广播接收器添加到ReceiverList中。
4. 广播发送过程

参考：<https://blog.csdn.net/luoshengyang/article/details/6744448>

1. 通过ContextWrapper.sendBroadcast()并最终调用ContextImpl. sendBroadcast ()；
2. ContextImpl. sendBroadcast ()调用ActivityManagerProxy.broadcastIntent将广播接收器的intent传给ActivityManagerServices，然后通过该intent找到相应的广播接收器；
3. ActivityManagerServices将接收到的广播放入自己的消息队列并串型处理（从这里可以看出广播的发送和处理是异步的）；
4. ActivityManagerServices调用通过binder调用ApplicationThread的scheduleRegisteredReceiver方法，将广播和广播接收器注册时创建的binder类型对象IntentReceiver分发给主进程；
5. ApplicaitonThread.scheduleRegisteredReceiver调用InnerReceiver.performReceive，并通过广播接收器注册时获取的handler将广播消息发送到ActivityThread的消息队列中；
6. ReceiverDispatcher通过其成员变量mReceiver（注册广播接收器时创建的BroadcastReceiver实例）的onReceive方法进行广播处理。