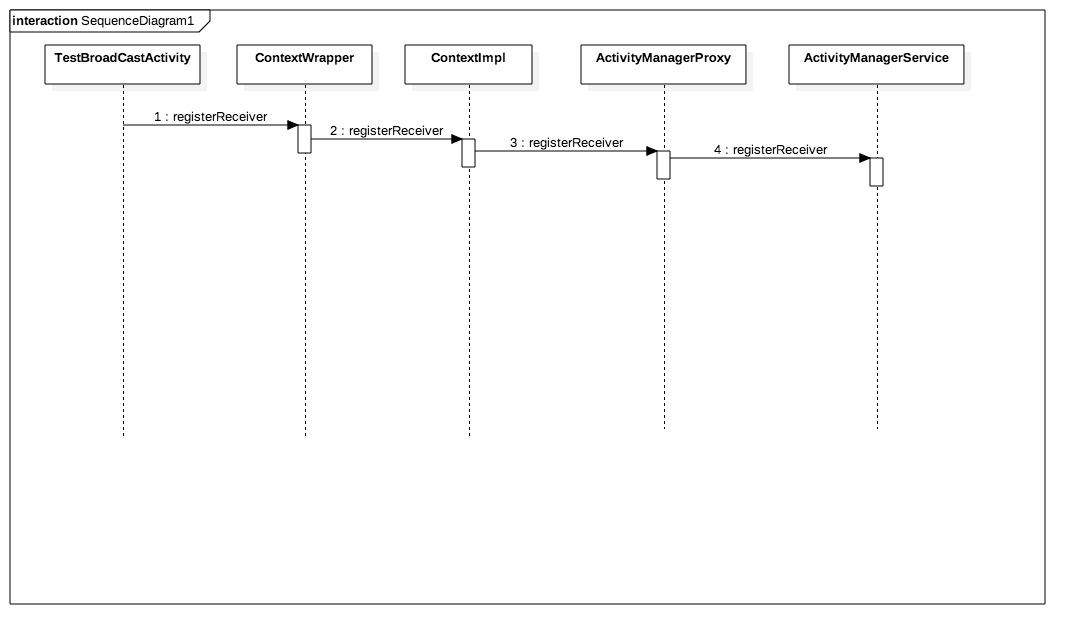
[插件占坑，四大组件动态注册前奏（三） 系统BroadCast的注册发送流程](http://blog.csdn.net/hejjunlin/article/details/52204143)

本节主要记录系统BroadCastReceiver的注册，发送流程：

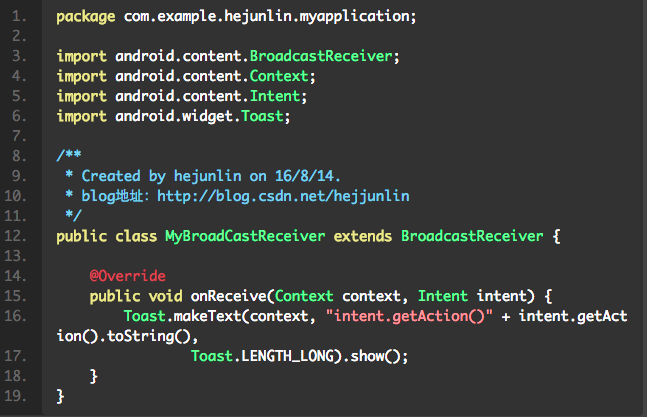
在了解注册，发送之前，先想一个问题：为什么有广播？

* **1、广播是一种组件之间传递的方式，这些组件可以运行在同一进程中，也可以运行在不同的进程中。**
* **2、广播的机制是建立在Binder进程间通信基础上的。在Binder进程间通信，Client组件在和Service组件进行通信之前，必须要先获取它的一个代理对象，即Client组件事先要知道Service组件的存在。然而，在广播机制中，广播发送者事先不需要知道广播接收者是存在的，这样就可以降低广播接收者和发送者之间的耦合度，得到模块分离。**
* **3、广播机制是一种基于消息发布和订阅的事件驱动模型，广播发送者负责发布消息，而广播接收者需要先订阅消息，然后才能收到消息。**
* **4、广播机制存在一个注册中心，它是由ActivityManagerService来担当的，广播接收者订阅消息的表现形式就是将自己注册到ActivieyManagerService中，并且指定要接收的广播的类型，当广播发送者向广播接收者发送一个广播时，这个广播首先发送到ActivityManagerService，然后ActivityManagerService可根据这个类型找到相应的广播接收者，最后将这个广播发送给它们处理。**
* **5、广播接收者的注册分为静态注册和动态注册，广播的发送方式分为有序和无序两种**
* **6、广播的生命周期，从对象调用它开始，到onReceiver方法执行完成之后结束。另外，每次广播被接收后会重新创建BroadcastReceiver对象，并在onReceiver方法中执行完就销毁，如果BroadcastReceiver的onReceiver方法中不能在10秒内执行完成，Android会出现ANR异常。所以不要在BroadcastReceiver的onReceiver方法中执行耗时的操作。**
* **7、如果需要在BroadcastReceiver中执行耗时的操作，可以通过Intent启动Service来完成。但不能绑定Service。特别是，您可能无法从一个BroadcastReceiver中显示一个对话框，或绑定到服务。对于前者，则应该使用NotificationManager的API。对于后者，你可以使用Context.startService()来启动一个Service。**

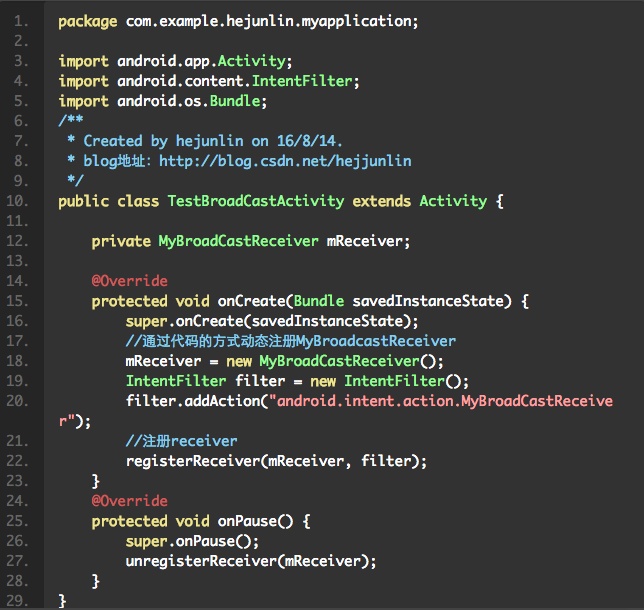
## BroadCastReceiver注册流程

先看一张时序图：   


首先得有一个广播类，以下叫MyBroadCastReceiver：



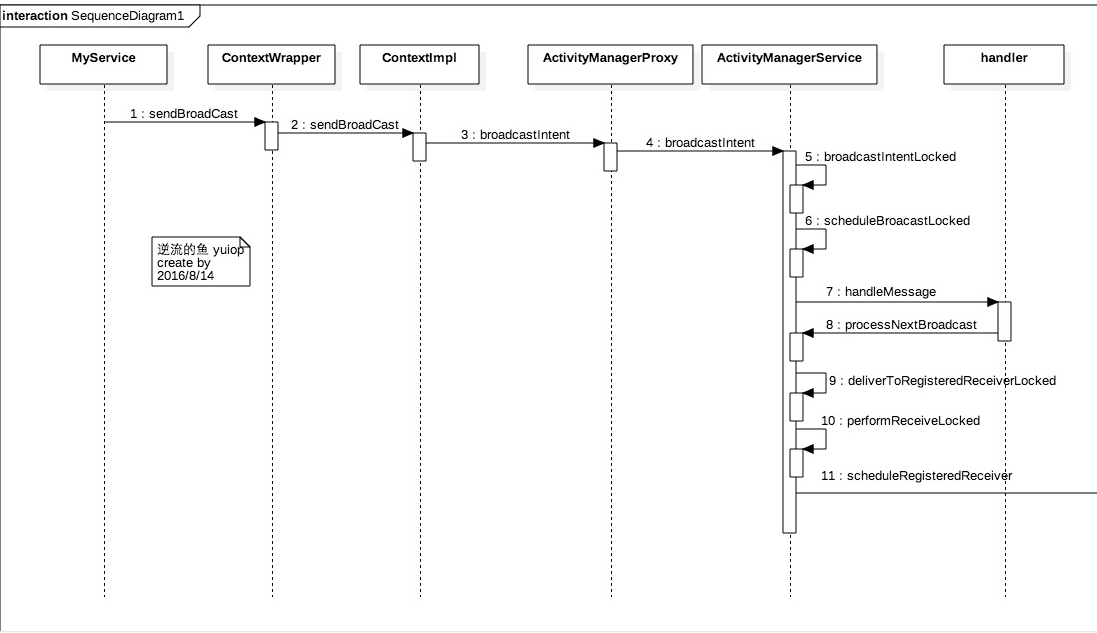
有一个TestBroadCastActivity的类：

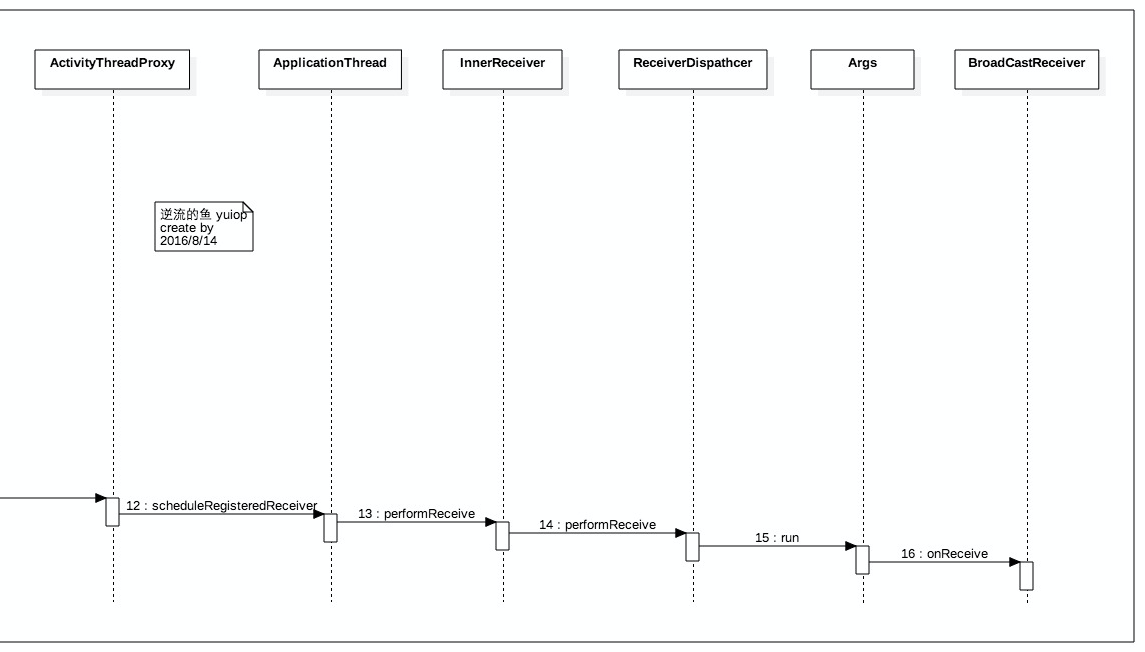


通过动态注册的方式注册了一个MyBroadCastReceiver广播类，以上就是一个广播的注册过程。

## BroadCastReceiver发送流程

先看下面两张时序图（ps：太长，只能分开截图）：发送过程远比注册过程复杂的多





广播的发送过程：

* **1、广播发送者，即一个Activity组件或者一个service组件，将一个特定类型的广播发送给ActivityManagerService;**
* 2 、ActivityManagerService接收到一个广播之后，首先找到与这个广播对应的广播接收者，然后将他们添加一个广播调度队列中，最后向   
  ActivityManagerService发送一个类型为BROADCAST\_INTENT\_MSG的值，这时广播对发送者来说，一个广播发送过程就完成了;
* **3、当发送到ActivityManagerService所运行的线程的消息队列中BROADCAST\_INTENT\_MSG消息被处理时，ActivityManagerService就会从广播调度队列中找到需要的广播的接收者，并且将对应的广播发送给它们所运行的应用程序进程。**
* **4、广播接收者所运行在的应用程序进程接收到ActivityManagerService发送过来的广播之后，并不是直接将接收到的广播分发给广播接收者来处理，而是将接收到的广播封装成一个消息，并且发送到主线程的消息队列中，当这个消息被处理时，应用程序进程才会将它所描述的广播发送给相应的广播接收者处理。**
* **5、ActivityManagerService向一个应用程序发送一个广播时，采用的是异步进程间通信,Binder驱动结构体中binder\_node时提到，发送给一个Binder实体对象的所有异步事务都是保存在一个异步事务队列中的，由于保存在一个异步事务队列中的异步事务在同一时刻只有一个会得到处理，即只有位于队列头部的异步事务才会得到处理，因为，ActivityManagerService就可以保证它发送给同一个应用程序的所有都可以按照发送顺序来串行地接收和处理。**

以上就是广播的注册发送过程，ContentProvider不再分析，下篇将正式进入插件占坑，四大组件动态化注册分析。