Service 与 BroadcaseReceiver

**本章要点**

* Service 组件的作用和意义
* 创建、配置Service
* 启动、停止Service
* 绑定本地Service并与之通信
* Service的生命周期
* 开发远程AIDL Service
* 在客户端程序中调用远程AIDL Service
* TelephonyManager的功能和用法
* 监听手机电话
* SmsManager的功能和用法
* 监听手机短信
* AudioManager的功能和用法
* BroadcastReceiver组件的作用和意义
* 开发、配置BroadcastReceiver组件
* 发送广播、发送有序广播
* 使用BroadcastReceiver接收系统广播

Service 是 Android四大组件中与Activity最相似的组件，它们都代表可执行的程序，Service与Activity 的区别在于：Service一直在后台运行，它没有用户界面，所以约不会到前台来。一早Service被启动起来之后，它就与Activity一样，完全具有自己的生命周期。关于程序中Activity与Service的选择标准是：如果某个程序组件需要在运行时向用户呈现某种界面，或者该程序需要与用户交互，就需要使用Activity，否则就应该考虑使用Service。

开发者开发Service的步骤与Activity的步骤很像，开发Service组件需要先开发一个Service的子类，然后在AndroidManifest.xml文件中配置该Service，配置时可通过<intent-filter …/>元素指定它可被哪些Intent启动。

Android系统本身提供了大量的Service组件，开发者可以通过这些系统Service来操作Android系统本身。

BroadcaseReceiver 组件就像是一个全局的事件监听器，只不过它用于监听系统发出来的Broadcast。通过使用BroadcastReceiver，即可在不同应用程序之间通信。

# 1 Service 简介

Service组件也是可执行的程序，它也有自己的生命周期。创建，配置Service与创建、配置Activity的过程基本相似。

## 1.1 创建、配置 Service

就像开发Activity需要两个步骤：

* 定义一个继承Service的子类
* 在AndroidManifest.xml文件中配置该Service。

Service与Activity还有一点相似之外，它们都是从Context派生出来的，因此它们都可以调用Context里定义的如 getResources(),getContextResolver()等方法。

与Activity相似的是，Service中也定义了一系列的生命周期方法。如下：

* abstract IBinder onBInd(Intent intent) : 该方法是Service子类必须实现的方法。该方法返回一个IBinder对象，应用程序可通过该对象与Service组件通信.
* void onCreate() : 当该Service第一次被创建后将立即架设该方法
* void onDestroy() : 当该Service被关闭之前将会回调该方法。
* void onStartCommand(Intent intent,int flags,int started):该方法的早期版本是void onStart(Intent intent,int started);每次客户端调用startService(intent) 方法启动该Service时都会架设方法。
* Boolean onUnBind(Intent intent) : 当该Service上绑定的所有客户端都断开连接时，将会架设方法方法。