

# Service

---

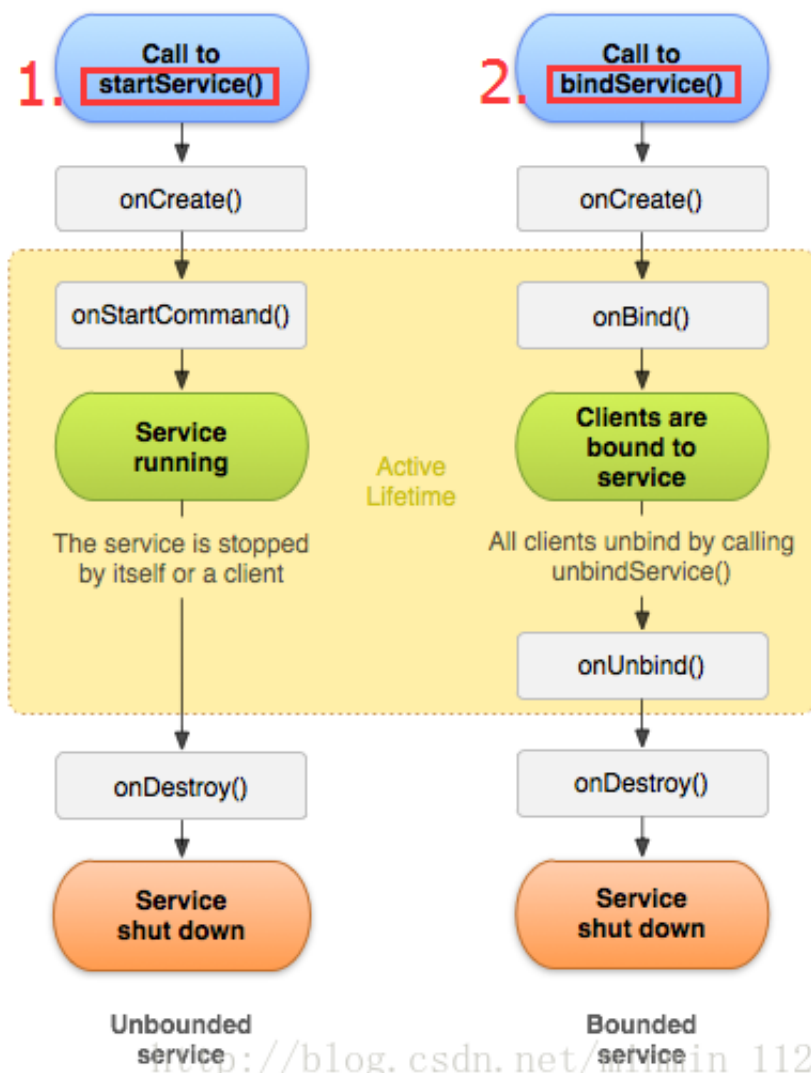
Android中实现程序后台运行的解决方案，它非常适合用于去执行那些不需要和用户交互而且还要求长期运行的任务。

- Service概要
- Service生命周期
- Service的基本用法
  - 普通Service
  - 前台Service
- IntentService
- Service与Activity的通信
- 如何保证Service不被杀死

## 1. Service概要

Service的运行不依赖于任何用户界面，因此即便程序被切换到后台或者用户打开了另一个应用程序，Service仍能够保持正常运行。但当某个应用程序进程被杀掉时，所有依赖于该进程的服务也会停止运行。实际上Service默认并不会运行在子线程中，也不运行在一个独立的进程中，它同样执行在主线程中（UI线程）。换句话说，不要在Service里执行耗时操作，除非你手动打开一个子线程，否则有可能出现主线程被阻塞（ANR）的情况

## 2. Service生命周期



回调方法含义：

**onCreate ()**：服务第一次被创建时调用

**onStartComand ()**：服务启动时调用

**onBind ()**：服务被绑定时调用

**onUnBind ()**：服务被解绑时调用

**onDestroy ()**：服务停止时调用

### **startService()启动**

此方法可以启动一个Service，并回调服务中的onStartCommand()。如果该服务之前还没创建，那么回调的顺序是onCreate()->onStartCommand()。服务启动了之后会一直保持运行状态，直到stopService()或stopSelf()方法被调用，服务停止并回调onDestroy()。另外，无论调用多少次startService()方法，只需调用一次stopService()或stopSelf()方法，服务就会停止了。

### **bindService()**

可以绑定一个Service，并回调服务中的onBind()方法。类似地，如果该服务之前还没创建，那么回调的顺序是onCreate()->onBind()。之后，调用方可以获取到onBind()方法里返回的IBinder对象的实例，从而实现和服务进行通信。只要调用方和服务之间的连接没有断开，服务就会一直保持运行状态，直到调用了unbindService()方法服务会停止，回调顺序onUnBind()->onDestroy()。

这两种启动方法并不冲突，当使用startService()启动Service之后，还可再使用bindService()绑定，只不过需要同时调用 stopService()和 unbindService()方法才能让服务销毁掉。

### 3.Service的基本用法

#### 普通Service

第一步：新建类并继承Service且必须重写onBind()方法，有选择的重写onCreate()、onStartCommand()及onDestroy()方法。

第二步：在配置文件中注册。学到现在会发现，四大组件除了广播接收器可用动态注册，定义好组件之后都要完成在配置文件注册的这一步。

第三步：在活动中利用Intent可实现Service的启动。

#### 前台Service

前台服务和普通服务最大的区别是，前者会一直有一个正在运行的图标在系统的状态栏显示，下拉状态栏后可以看到更加详细的信息，非常类似于通知的效果。使用前台服务或者为了防止服务被回收掉，比如听歌，或者由于特殊的需求，比如实时天气状况。想要实现一个前台服务非常简单，它和之前学过的发送一个通知非常类似，只不过在构建好一个Notification之后，不需要NotificationManager将通知显示出来，而是调用了startForeground()方法。

### 4.IntentService

可以简单地创建一个异步的、会自动停止的服务

#### 使用步骤

第一步：新建类并继承IntentService，在这里需要提供一个无参的构造函数且必须在其内部调用父类的有参构造函数，然后具体实现 onHandleIntent()方法，在里可以去处理一些耗时操作而不用担心 ANR的问题，因为这个方法已经是在子线程中运行的了。第二步：在配置文件中注册。第三步：在活动中利用Intent实现IntentService的启动，和Service用的方法是完全一样的。

### 5.Service与Activity的通信

借助服务的onBind()方法

### 6.如何保证Service不被杀死

- 技术点：Service应用
- 思路：列举几种解决办法
- 参考回答：可以采取以下几种解决方法：

在Service的onStartCommand()中设置flags值为START\_STICKY，使得Service被杀死后尝试再次启动Service

提升Service优先级，比如设置为一个前台服务

在Activity的onDestroy()通过发送广播，并在广播接收器的onReceive()中启动Service