# Android開發四大組件---Service詳解

# <http://android.tgbus.com/Android/androidnews/201204/419662.shtml>

service可以在和多場合的應用中使用，比如播放多媒體的時候用戶啟動了其他Activity這個時候程序要在後台繼續播放，比如檢測SD卡上文件的變化，再或者在後台記錄你地理信息位置的改變等等，總之服務嘛，總是藏在後頭的。

　　Service是在一段不定的時間運行在後台，不和用戶交互應用組件。每個Service必須在manifest中通過<service>來聲明。可以通過contect.startservice和contect.bindserverice來啟動。

　　Service和其他的應用組件一樣，運行在進程的主線程中。這就是說如果service需要很多耗時或者阻塞的操作，需要在其子線程中實現。

　　service的兩種模式(startService()/bindService()不是完全分離的)：

　　本地服務Local Service 用於應用程序內部。

　　它可以啟動並運行，直至有人停止了它或它自己停止。在這種方式下，它以調用Context.startService()啟動，而以調用Context.stopService()結束。它可以調用Service.stopSelf() 或Service.stopSelfResult()來自己停止。不論調用了多少次startService()方法，你只需要調用一次stopService()來停止服務。

　　用於實現應用程序自己的一些耗時任務，比如查詢升級信息，並不佔用應用程序比如Activity所屬線程，而是單開線程後台執行，這樣用戶體驗比較好。

　　遠程服務Remote Service 用於android系統內部的應用程序之間。

　　它可以通過自己定義並暴露出來的接口進行程序操作。客戶端建立一個到服務對象的連接，並通過那個連接來調用服務。連接以調用Context.bindService()方法建立，以調用Context.unbindService()關閉。多個客戶端可以綁定至同一個服務。如果服務此時還沒有加載，bindService()會先加載它。

　　可被其他應用程序復用，比如天氣預報服務，其他應用程序不需要再寫這樣的服務，調用已有的即可。

　　生命週期

　　使用context.startService() 啟動Service是會會經歷:

　　context.startService() ->onCreate()- >onStart()->Service running

　　context.stopService() | ->onDestroy() ->Service stop

　　如果Service還沒有運行，則android先調用onCreate()然後調用onStart();如果Service已經運行，則只調用onStart()，所以一個Service的onStart方法可能會重複調用多次。

　　stopService的時候直接onDestroy，如果是調用者自己直接退出而沒有調用stopService的話，Service會一直在後台運行。該Service的調用者再啟動起來後可以通過stopService關閉Service。

　　所以調用startService的生命週期為：onCreate --> onStart(可多次調用) --> onDestroy

　　使用使用context.bindService()啟動Service會經歷：

　　context.bindService()->onCreate()->onBind()->Service running

　　onUnbind() -> onDestroy() ->Service stop

　　onBind將返回給客戶端一個IBind接口實例，IBind允許客戶端回調服務的方法，比如得到Service運行的狀態或其他操作。這個時候把調用者(Context，例如Activity)會和Service綁定在一起，Context退出了，Srevice就會調用onUnbind->onDestroy相應退出。

　　所以調用bindService的生命週期為：onCreate --> onBind(只一次，不可多次綁定) --> onUnbind --> onDestory。

　　在Service每一次的開啟關閉過程中，只有onStart可被多次調用(通過多次startService調用)，其他onCreate，onBind，onUnbind，onDestory在一個生命週期中只能被調用一次。

　　而啟動service，根據onStartCommand的返回值不同，有兩個附加的模式：

　　1. START\_STICKY 用於顯示啟動和停止service。

　　2. START\_NOT\_STICKY或START\_REDELIVER\_INTENT用於有命令需要處理時才運行的模式。

　　服務不能自己運行，需要通過調用Context.startService()或Context.bindService()方法啟動服務。這兩個方法都可以啟動Service，但是它們的使用場合有所不同。

　　1. 使用startService()方法啟用服務，調用者與服務之間沒有關連，即使調用者退出了，服務仍然運行。

　　如果打算採用Context.startService()方法啟動服務，在服務未被創建時，系統會先調用服務的onCreate()方法，接著調用onStart()方法。

　　如果調用startService()方法前服務已經被創建，多次調用startService()方法並不會導致多次創建服務，但會導致多次調用onStart()方法。

　　採用startService()方法啟動的服務，只能調用Context.stopService()方法結束服務，服務結束時會調用onDestroy()方法。

　　2. 使用bindService()方法啟用服務，調用者與服務綁定在了一起，調用者一旦退出，服務也就終止，大有“不求同時生，必須同時死”的特點。

　　onBind()只有採用Context.bindService()方法啟動服務時才會回調該方法。該方法在調用者與服務綁定時被調用，當調用者與服務已經綁定，多次調用Context.bindService()方法並不會導致該方法被多次調用。

　　採用Context.bindService()方法啟動服務時只能調用onUnbind()方法解除調用者與服務解除，服務結束時會調用onDestroy()方法。