

Hướng dẫn lập trình cơ bản với Android

Part 18

Android Service

4 Tutorial trước các bạn đã có 1 lượng kiến thức kha khá, tiếp sau đây mình xin giới thiệu 1 khái niệm cơ bản nữa trong android, đó là Service

Service là 1 trong 4 thành phần chính trong 1 ứng dụng Android (Activity, Service, BroadcastReceiver, ContentProvider) thành phần này chạy trong hậu trường và làm những công việc không cần tới giao diện như chơi nhạc, download, xử lý tính toán...

Một Service có thể được sử dụng theo 2 cách:

- Nó có thể được bắt đầu và được cho phép hoạt động cho đến khi một người nào đó dừng nó lại hoặc nó tự ngắt. Ở chế độ này, nó được bắt đầu bằng cách gọi `Context.startService()` và dừng bằng lệnh `Context.stopService()`. Nó có thể tự ngắt bằng lệnh `Service.stopSelf()` hoặc `Service.stopSelfResult()`. Chỉ cần một lệnh `stopService()` để ngừng Service lại cho dù lệnh `startService()` được gọi ra bao nhiêu lần

- Service có thể được vận hành theo như đã được lập trình việc sử dụng một Interface mà nó định nghĩa. Các người dùng thiết lập một đường truyền tới đối tượng Service và sử dụng đường kết nối đó để thâm nhập vào Service. Kết nối này được thiết lập bằng cách gọi lệnh `Context.bindService()` và được đóng lại bằng cách gọi lệnh `Context.unbindService()`. Nhiều người dùng có thể kết nối tới cùng một thiết bị. Nếu Service vẫn chưa được khởi chạy, lệnh `bindService()` có thể tùy ý khởi chạy nó. Hai chế độ này thì không tách biệt toàn bộ. Bạn có thể kết nối với một Service mà nó đã được bắt đầu với lệnh `startService()`. Ví dụ, một Service nghe nhạc ở chế độ nền có thể được bắt đầu bằng cách gọi lệnh `startService()` cùng

với một đối tượng Intent mà định dạng được âm nhạc để chơi. Chỉ sau đó, có thể là khi người sử dụng muốn kiểm soát trình chơi nhạc hoặc biết thêm thông tin về bài hát hiện tại đang chơi, thì sẽ có một Activity tạo lập một đường truyền tới Service bằng cách gọi `bindService()`. Trong trường hợp như thế này, `stopService()` sẽ không thực sự ngừng Service cho đến khi liên kết cuối cùng được đóng lại.

Giống như một Activity, một Service cũng có các phương thức chu kỳ thời gian mà bạn có thể cài đặt để kiểm soát những sự thay đổi trong trạng thái của nó. Những phương thức của Service thì ít hơn là của Activity – chỉ có 3- và chúng thì được sử dụng rộng rãi, không được bảo vệ.

`void onCreate()`

`void onStart(Intent intent)`

`void onDestroy()`

Bằng việc thực hiện những phương thức này, bạn có thể giám sát 2 vòng lặp của chu kỳ thời gian của mỗi Service

Entire lifetime của một Service diễn ra giữa thời gian `onCreate()` được gọi ra và thời gian mà `onDestroy()` trả lại. Giống như một Activity, một Service lại tiết hành cài đặt ban đầu ở `onCreate()`, và giải phóng tất cả các tài nguyên còn lại ở `onDestroy()` Ví dụ, một Service phát lại nhạc có thể tạo ra một luồng và bắt đầu chơi nhạc `onCreate()`, và sau đó luồng chơi nhạc sẽ dừng lại ở `onCreate()`,

Active lifetime của một Service bắt đầu bằng một lệnh tới `onStart()`. Đây là phương thức được chuyển giao đối tượng Intent mà đã được thông qua để tới `startService()` Service âm nhạc sẽ mở đối tượng Intent để quyết định xem sẽ chơi loại nhạc nào và bắt đầu phát nhạc.

Không có callback tương đương nào cho thời điểm Service ngừng lại – không có phương thức `onStop()`

Các phương thức `onCreate()` và `onDestroy()` được gọi cho tất cả các Service dù chúng có được bắt đầu bằng `Context.startService()` hoặc `Context.bindService()` hay không. Tuy nhiên thì, `onStart()` chỉ được gọi ra đối với các Service bắt đầu bằng `startService()`.

Nếu một Service cho phép những Service khác kết nối với nó thì sẽ có thêm các phương thức callback dành cho Service đó để thực hiện

`IBinder onBind(Intent intent)`

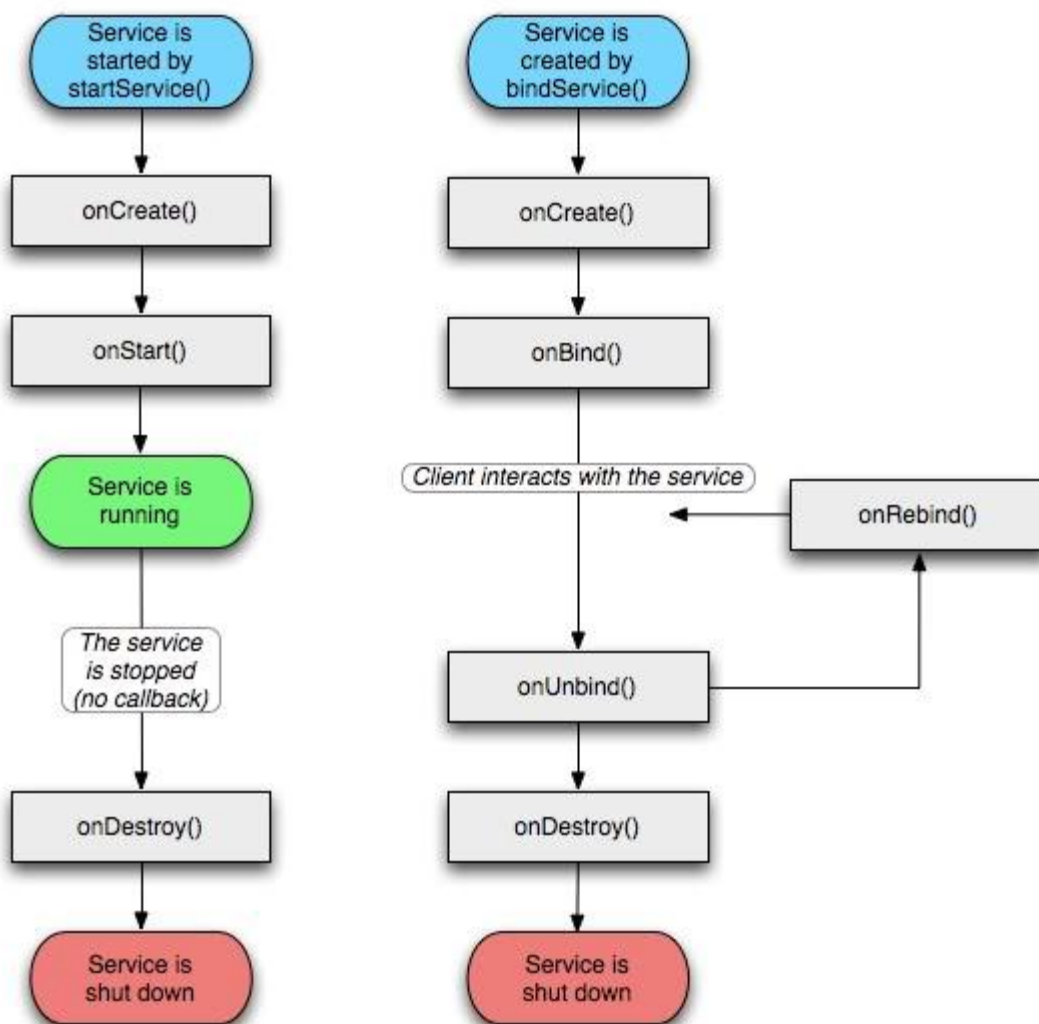
`boolean onUnbind(Intent intent)`

`void onRebind(Intent intent)`

Hàm callback `onBind()` thông qua đối tượng Intent đã được truyền đến `bindService` và `onUnbind()` được chuyển giao đối tượng mà đã được chuyển đến. Nếu Service

đang được chỉ định (binding), onBind() quay trở lại kênh thông tin mà người dùng sử dụng để tương tác với Service. Phương thức onUnbind() có thể yêu cầu onRebind() được gọi nếu một người dùng kết nối với Service

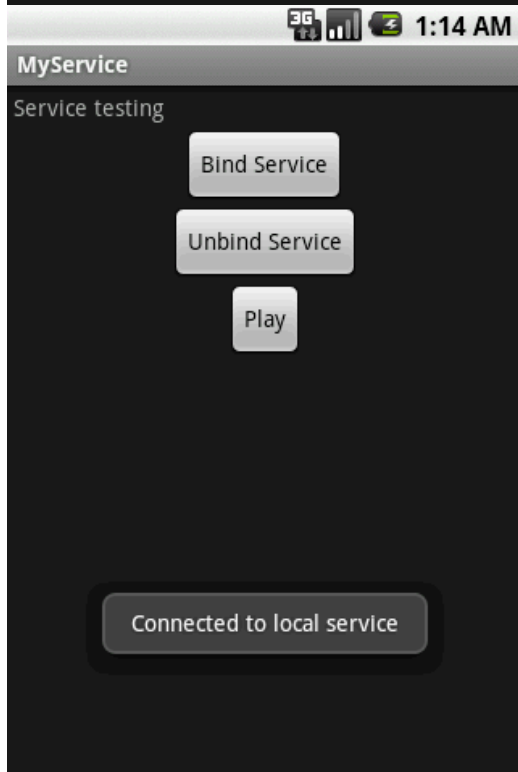
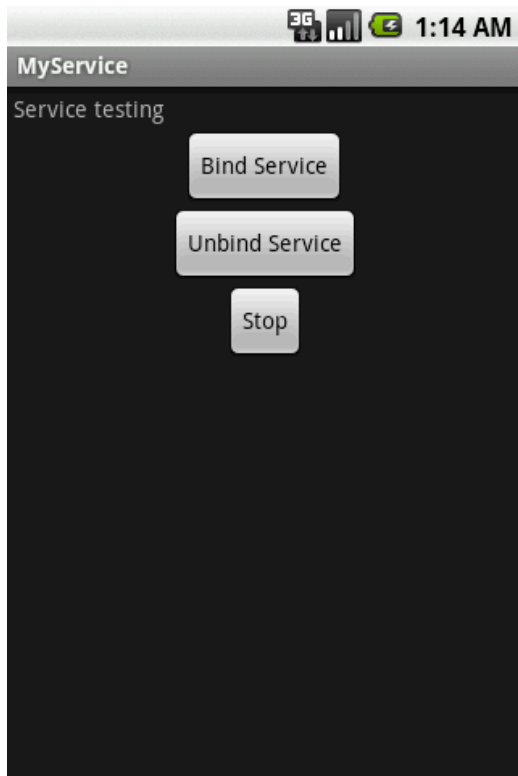
Biểu đồ dưới đây minh họa cho các phương thức callback giành cho một Service. Mặc dù, nó phân tách các Service được tạo ra thông qua startService với các Service mà được tạo ra bằng bindService(). Hãy nhớ rằng bất kì Service nào, cho dù nó được khởi tạo như thế nào thì nó vẫn có thể cho phép các người dùng kết nối tới nó một cách hiệu quả nhất, cho nên bất kì Service nào cũng có thể được chỉ định thông qua các các phương thức onBind() và onUnbind()



Service LifeCycle

Các bạn đã đọc và hiểu về Service trong Part 1. Tiếp theo mình sẽ làm 1 demo nhỏ để các bạn hiểu rõ hơn về Service. Demo tạo 1 service chơi nhạc và người sử dụng có thể điều khiển service này.

Giao diện chương trình :



MyService

Service testing

Bind Service

Unbind Service

Play