BroadcastReceiver

MultiUni
Trần Vũ Tất Bình

Tổng quan

- BroadcastReceiver (có thể gọi là Receiver là một trong bốn loại thành phần trong ứng dụng Android. Chức năng dùng để nhận các sự kiện mà các ứng dụng hoặc hệ thống phát đi.
- Có 2 cách phát-nhận đó là:
  - Không có thứ tự: receiver nào đủ điều kiện thì nhận hết, không phân biệt và cũng tách rời nhau.
  - Có thứ tự: receiver nào đăng ký ưu tiên hơn thì nhận trước, và có thể truyền thêm thông tin xử lý cho các receiver sau.

Lifecycle

- Thực ra lifecycle của BroadcastReceiver chỉ có duy nhất một phương thức onReceive().
  - Khi có sự kiện mà BroadcastReceiver đã đăng ký nhận được phát đi, thì phương thức onReceive() của BroadcastReceiver đó sẽ được gọi.
  - Sau khi thực thi xong phương thức này, lifercycle của Receiver kết thúc.

Lưu ý khi sử dụng

- Ngay khi onReceive() kết thúc, hệ thống coi như receiver đã không còn hoạt động và có thể kill process chứa receiver này bất cứ lúc nào.

  →Tránh xử lý các code quá lâu trong onReceive().

  → Không có xử lý bất đồng bộ, chờ callback… trong Receiver (cụ thể như hiển thị Dialog, kết nối service…)

Một số broadcast thông dụng

- Báo hệ thống khởi động xong
- Báo pin có sự thay đổi
- Báo có package mới cài vào hoặc xóa đi
- Báo tắt máy
- Báo cắm sạc, rút sạc…

  Xem thêm tại đây, mục Standard Broadcast Actions

Một số broadcast khác

- Thông báo tin nhắn tới
- Thông báo cắm, rút thẻ nhớ
- Thông báo có cuộc gọi đi
- Và các bạn có thể định nghĩa broadcast cho riêng mình (mục tiêu chính của việc này giúp bạn có thể liên lạc giữa các ứng dụng bạn viết hoặc thông báo một sự kiện liên quan đến ứng dụng của bạn với các ứng dụng khác)

onReceive()

- Phương thức này được gọi khi có sự kiện tương ứng được phát đi. Ở trong phương thức này, ta thấy truyền vào context và intent.
  - Vì Receiver không kế thừa từ Context nên cần truyền context mà receiver này đang chạy vào. Thứ nhất, để có thể xử lý các phương thức yêu cầu truyền thêm Context, thứ 2, để sử dụng các phương thức của lớp Context. (còn nữa hay không thì các bạn giúp mình luôn nhé)

onReceive()

- Intent được truyền vào sẽ có đầy đủ thông tin như sự kiện nào mà receiver này đăng ký đã xảy ra dẫn đến onReceive() được gọi. Có gửi kèm thông tin gì hoặc dữ liệu gì hay không. Xem các api:

  Intent.getAction()

  Intent.get…Extra(String dataName)

Ví dụ BootReceiver

- Các bạn có thể đăng ký nhận sự kiện hệ thống vừa khởi động xong để có thể làm việc gì đó ngay, hoặc vận hành song song với hệ thống…
- Ta sẽ đăng ký nhận sự kiện BOOT_COMPLETED, sau đó sẽ gọi một dialog lên hiển thi lời chào.
- Khi hệ thống khởi động xong sẽ xuất một dialog chào user ☺

Ví dụ BootReceiver

- Trong manifest, cần đăng ký permission được nhận sự kiện này

```
<uses-permission
    android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```

- Khai báo receiver bên trong thẻ application

```
<receiver android:name=".BootReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>
```

Ví dụ BootReceiver

- Ở đây ta khai báo trong manifest là ứng dụng có một receiver tên là BootReceiver (slide kế sẽ thấy).
- BootReceiver này đăng ký nhận sự kiện “hệ thống khởi động hoàn tất”.
- Dĩ nhiên, muốn nhận sựu kiện dạng này thì cần phải đăng kí trước để với hệ thống qua permission để user được biết.

Ví dụ BootReceiver

- Tạo một class mới trong source, tên là BootReceiver kế thừa BroadcastReceiver.
- Implement lại phương thức onReceive() với code như sau:

```
if (intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED)) {
    Intent helloIntent = new Intent(context, HelloBootActivity.class);
    helloIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(helloIntent);
}
```

Ví dụ BootReceiver

- Giải thích một chút: vì Receiver không kế thừa context nên khi tạo intent mới không truyền this vô được, thay vào đó truyền cái context đã được gửi kèm.
- Vì không ở trong 1 activity mà đang ở trong 1 receiver, và một số vấn đề liên quan tới task trong Android nên bạn phải thêm cờ Intent.*FLAG_ACTIVITY_NEW_TASK* (bạn chỉ có thể ko dùng cờ này khi bạn gọi startActivity() từ một activity)

Ví dụ BootReceiver

- Đọc tới đây chắc các bạn hiểu ngay là ứng dụng có một activity tên là HelloBootActivity, activity này sẽ chỉ hiển thị dạng dialog, và sẽ không được start bằng cách bấm vào icon trên màn hình. Vì thế, khai báo trong manifest như sau:

```
<activity android:name=".HelloBootActivity"
        android:theme="@android:style/Theme.Dialog">
</activity>
```

- Còn activity chỉ hiển thị một cái TextView là "Chào bạn, mới khởi động xong" và một cái Button để bấm vào đó thì đóng activity, các bạn hoàn tất nhé.

Phát sự kiện

- Các bạn có thể phát một sự kiện cho các receiver khác nhận dạng như sau:

```
Intent intent = new
        Intent("org.multiuni.android.BROADCAST_DEMO");
sendBroadcast(intent);
```

- Hoặc:

```
sendOrderedBroadcast(intent, "permission tùy ý hoặc null");
```

- Ngoài ra còn có một số các gửi broadcast khác, các bạn tham khảo thêm trong class ContextWrapper nhé

Bài tập yêu cầu

1. Làm ứng dụn BootReceiver vừa rồi.
2. Lên g-android, tìm code mẫu về chuyển hướng cuộc gọi, làm cái đó (cũng receiver)
3. Viết 2 ứng dụng, ứng dụng A nhập vào một chuỗi rồi phát đi một sự kiện tự định nghĩa, kèm theo chuỗi đó. Ứng dụng B nhận sự kiện đó và hiển thị một Toast có nội dung là chuỗi nhận được.