

Hướng dẫn lập trình cơ bản với Android

Part 13

Trong bài này mình sẽ đi sâu nói rõ về Intent, phần cơ bản và đóng vai trò rất quan trọng trong lập trình ứng dụng Android.

Khái niệm về Intent:

Theo định nghĩa của Google, Intent là một miêu tả về một hoạt động cần được thực hiện. Còn nói một cách đơn giản và dễ hiểu hơn, Intent là một cơ cấu cho phép truyền thông điệp giữa các thành phần của 1 ứng dụng và giữa các ứng dụng với nhau.

Các thuộc tính của Intent:

- **action**: là hành động được thực hiện, vd : ACTION_VIEW, ACTION_MAIN
- **data**: là dữ liệu sẽ được xử lý trong action, thường được diễn tả là một Uri (Uniform Resource Identifier, tham khảo http://en.wikipedia.org/wiki/Uniform...rce_Identifier để hiểu rõ thêm chi tiết).

VD:

ACTION_VIEW content://contacts/people/1 - Hiển thị thông tin về người với mã danh 1

ACTION_DIAL content://contacts/people/1 - Hiển thị màn hình gọi đến người với mã danh 1

ACTION_DIAL tel:123 - Hiển thị màn hình gọi với số gọi là 123

Ngoài ra còn có 1 số thuộc tính mà ta có thể bổ sung vào Intent:

- **category**: bổ sung thêm thông tin cho action của Intent. VD:

CATEGORY_LAUNCHER thông báo sẽ thêm vào Launcher như là một ứng dụng

top-level

- **type**: chỉ rõ kiểu của data
- **component**: chỉ rõ thành phần sẽ nhận và xử lý intent. Khi thuộc tính này được xác định thì các thuộc tính khác sẽ trở thành thuộc tính phụ.
- **extras**: mang theo đối tượng Bundle chứa các giá trị bổ sung.

VD:

ACTION_MAIN và **CATEGORY_HOME**: trở về màn hình Home của Android (khi bấm nút Home của di động)

Phân loại Intent:

Intent được chia làm 2 loại:

- **Explicit Intents**: intent đã được xác định thuộc tính component, nghĩa là đã chỉ rõ thành phần sẽ nhận và xử lý intent. Thông thường intent dạng này sẽ không bổ sung thêm các thuộc tính khác như action, data. Explicit Intent thường được sử dụng để khởi chạy các activity trong cùng 1 ứng dụng.
- **Implicit Intents**: Intent không chỉ rõ component xử lý, thay vào đó nó bổ sung thông tin trong các thuộc tính. Khi intent được gửi đi, hệ thống sẽ dựa vào những thông tin này để quyết định component nào thích hợp nhất để xử lý nó.

VD:

ACTION_DIAL tel:123 thông thường sẽ được hệ thống giao cho activity Phone Dialer mặc định của Android xử lý.

Một số action thường sử dụng trong Intent:

ACTION_ANSWER - mở Activity để xử lý cuộc gọi tới, thường là Phone Dialer của Android

ACTION_CALL - mở 1 Phone Dialer (mặc định là PD của Android) và ngay lập tức thực hiện cuộc gọi dựa vào thông tin trong data URI

ACTION_DELETE - mở Activity cho phép xóa dữ liệu mà địa chỉ của nó chứa trong data URI

ACTION_DIAL - mở 1 Phone Dialer (mặc định là PD của Android) và điền thông tin lấy từ địa chỉ chứa trong data URI

ACTION_EDIT - mở 1 Activity cho phép chỉnh sửa dữ liệu mà địa chỉ lấy từ data URI

ACTION_SEND - mở 1 Activity cho phép gửi dữ liệu lấy từ data URI, kiểu của dữ liệu xác định trong thuộc tính type

ACTION_SENDTO - mở 1 Activity cho phép gửi thông điệp tới địa chỉ lấy từ data URI

ACTION_VIEW - action thông dụng nhất, khởi chạy activity thích hợp để hiển thị dữ liệu trong data URI

ACTION_MAIN - sử dụng để khởi chạy 1 Activity

OK, lý thuyết như thế là đã tạm ổn. Giờ chúng ta sẽ chuyển qua phần thực hành để hiểu rõ cách sử dụng Intent. Như đã nêu ở trên, Intent chia làm 2 loại: explicit intent và implicit intent. Mỗi loại Intent sẽ có cách cài đặt và sử dụng khác nhau.

Using Explicit Intents

Yêu cầu: Xây dựng chương trình gồm 2 Activity. Activity1 là Activity chạy ban đầu lúc khởi động ứng dụng, cho phép nhập vào 1 giá trị, cho phép khởi chạy Activity2 và gửi giá trị này tới Activity2. Activity2 sẽ nhận và hiển thị giá trị, rồi lại gửi giá trị này tới 1 BroadcastReceiver. Cơ chế gửi và khởi chạy Activity sử dụng thông qua Intent.

B1: Khởi tạo project: File -> New -> Android Project

Project name: Explicit Intent Example

Build Target: Chọn Android 1.5

Application name: Explicit Intent Example

Package name: at.exam

Create Activity: Activity1

=> Kích nút Finish.

B2: Tạo giao diện cho Activity1 -> res\layout\main.xml chuyển tên thành activity1_layout.xml

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Activity 1 - Send value"
        android:typeface="normal"
        android:textSize="14px"
        android:textStyle="bold"
        android:textColor="#cccccc"
        android:background="#333333"
    />
    <EditText
        android:id="@+id/value_edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:gravity="center"
        android:lines="1"
        android:numeric="integer"
    />
    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <Button
            android:id="@+id/send_button"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Send to Activity 2"
            android:layout_alignParentBottom="true"
        />
    </RelativeLayout>
</LinearLayout>
```

Layout cho Activity1 bao gồm 1 LinearLayout chứa 1 TextView, 1 EditText để nhập giá trị (đã giới hạn kiểu nhập là number), và 1 RelativeLayout có 1 Button để khởi chạy Activity2. Mình sử dụng RelativeLayout để có thể xếp Button này xuống phía cuối của giao diện.