

Hướng dẫn lập trình cơ bản với Android

Part 21

Android Content Provider

Cuối tuần rảnh rồi hơn, nhớ ra là phải tiếp tục viết Tutorial cho diễn đàn, rất nhiều các Newbies đang cần 🙏

Trước khi vào bài, các bạn down Sourcecode của bài học [tại đây](#)

Sourcecode của bài học bao gồm 3 Project demo :

ContentProviderDemo

ContentProviderDemo1

ContentProviderDemo2

Mỗi Demo sẽ giải quyết từng vấn đề của Tutorial. Các bạn cứ add hết vào workspace để tiện theo dõi.

1. Giới thiệu Content Provider

Content Provider là 1 trong 4 thành phần cơ bản của 1 ứng dụng Android thường có bao gồm:

1. Activity
2. Service
3. Broadcast Receiver
4. Content Provider

Một Content Provider cung cấp một tập chi tiết dữ liệu ứng dụng đến các ứng dụng khác. Thường được sử dụng khi chúng ta muốn tạo cơ sở dữ liệu dưới dạng public (các ứng dụng khác có thể truy xuất).

Dữ liệu thường được lưu trữ ở file hệ thống, hoặc trong một SQLite database. Đơn giản để các bạn có thể hình dung như : Danh bạ, Call log, cấu hình cài đặt...trên điện thoại là dữ liệu dưới dạng Content Provider.

Content Provider hiện thực một tập phương thức chuẩn mà các ứng dụng khác có thể truy xuất và lưu trữ dữ liệu của loại nó điều khiển.

Tuy nhiên, những ứng dụng không thể gọi các phương thức trực tiếp. Hơn thế chúng dùng lớp Content Resolver và gọi những phương thức đó. Một Content Resolver có thể giao tiếp đến nhiều content provider; nó cộng tác với các provider để quản lý bất kỳ giao tiếp bên trong liên quan.

Đơn giản hơn, chúng ta có thể làm 1 ứng dụng nhỏ để lấy tất cả các thông tin cấu hình trong máy load lên listview. Các bạn có thể chạy Project ContentProviderDemo1 trong SourcecodeDemo.



Content Provider Demo	
volume_music	11
volume_system	5
volume_voice	4
volume_alarm	6
mode_ringer	2
vibrate_on	4
mode_ringer_streams_affected	38
mute_streams_affected	46
dim_screen	1
stay_on_while_plugged_in	1
screen_off_timeout	60000
emergency_tone	0



Content Provider Demo	
airplane_mode_radicell,bluetooth,wifi os	
auto_time	1
screen_brightness	102
window_animation_scale	1.0
transition_animation_scale	0.0
accelerometer_rotation	1
volume_notification	5
volume_notification_last_audible	5
volume_ring	5
volume_ring_last_audible	5
next_alarm_formatted	
font_scale	1.0

Chúng ta có thể tìm hiểu sơ qua về code của demo này, rất ngắn gọn

Mã:

```
        ContentResolver cr = getContentResolver();
        Cursor cursor =
cr.query(Settings.System.CONTENT_URI, null, null, null,
null);
        startManagingCursor(cursor);

        ListView listView = (ListView)
findViewById(R.id.listView);
        String[] from = { Settings.System.NAME,
Settings.System.VALUE };
        int[] to = { R.id.textName, R.id.textValue };
        SimpleCursorAdapter adapter = new
SimpleCursorAdapter(this, R.layout.row, cursor, from,
to);
        listView.setAdapter(adapter);
```

Như các bạn thấy, chỉ cần 2 dòng code đơn giản để lấy được con trỏ thao tác trên tập dữ liệu cần lấy:

Mã:

```
        ContentResolver cr = getContentResolver();
        Cursor cursor =
cr.query(Settings.System.CONTENT_URI, null, null, null,
null);
```

Lớp Content Resolver cung cấp các phương thức xử lý dữ liệu thông qua các Uri, mỗi Content Provider có 1 Uri cụ thể, ở đây Uri *Settings.System.CONTENT_URI* sẽ trả lại tập dữ liệu là thông tin cấu hình của thiết bị.

Sau khi lấy được con trỏ tới tập dữ liệu, việc còn lại đơn giản là bind data lên listview để hiển thị:

Mã:

```
        startManagingCursor(cursor);
        ListView listView = (ListView)
findViewById(R.id.listView);
        String[] from = { Settings.System.NAME,
Settings.System.VALUE };
        int[] to = { R.id.textName, R.id.textValue };
        SimpleCursorAdapter adapter = new
SimpleCursorAdapter(this, R.layout.row, cursor, from,
to);
        listView.setAdapter(adapter)
```