

Bài 2: Các thành phần cơ bản của ứng dụng Android

# LẬP TRÌNH ANDROID CƠ BẢN

**Link videos:**

<https://www.youtube.com/playlist?list=PL0YqB-plTBzuSdqePxDnKQfoSZpoPKhzw>

## Nội dung bài học

- Intent, Content
  - Activity và vòng đời của Activity
  - Service và vòng đời của Service
  - Tài nguyên ứng dụng
- <https://www.youtube.com/@LapTrinhJava/playlists>

## Website tham khảo

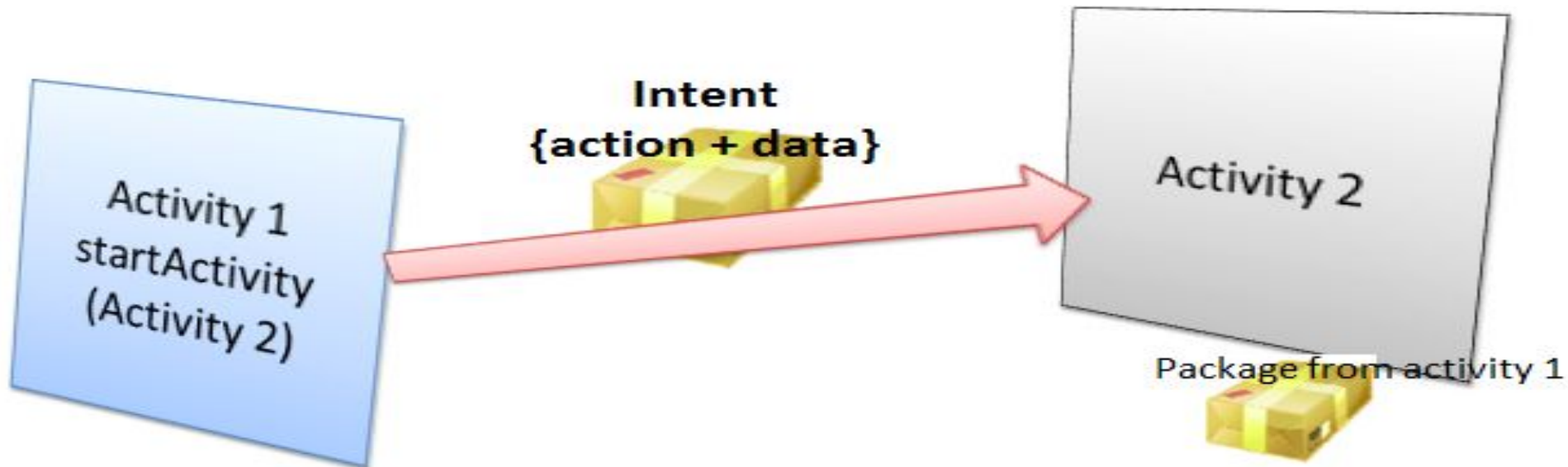
- Android Developers Blog  
<http://android-developers.blogspot.com/?hl=en>
- Android Central  
<http://www.androidcentral.com/>

- Các thành phần:
  - Activity
  - Service
  - ContentProvider
  - BroadcastReceiver
- Intent:
  - Action, Data = Implicit
  - Action, Data, Component = Explicit



## Thành phần khác của Intent

- Category: miêu tả loại thành phần điều khiển Intent
  - CATEGORY\_LAUNCHER: Activity xuất hiện ở launcher
  - CATEGORY\_PREFERENCE
- EXTRA: cặp giá trị key-value chứa thông tin bổ sung
  - ACTION\_HEADSET\_PLUG
- Flags: hướng dẫn hệ thống cách khởi tạo Activity
  - FLAG\_ACTIVITY\_NO\_ANIMATION



## Truy cập các thành phần ứng dụng

- Activity và service được khởi tạo như thế nào?
- Java:
  - Viết class để thực hiện một số công việc
    - Giống trong Android
  - Viết phương thức Main. Trong phương thức Main gọi hàm khởi tạo của class và chạy các phương thức
    - Không giống trong Android
    - Phụ thuộc vào kiểu đối tượng, Android sẽ gọi hàm tạo và quản lý vòng đời của đối tượng

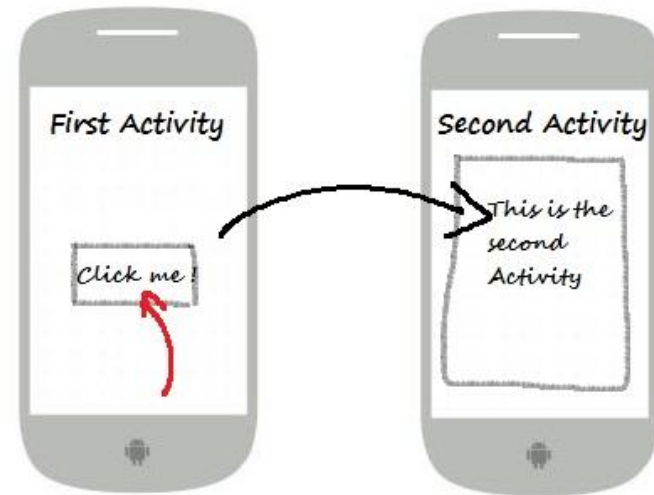
# Context

- Lớp Context cung cấp truy cập tới chức năng và dịch vụ của hệ thống
- Activity và Service kế thừa Context, do đó có thể gọi các phương thức trong Context trực tiếp
- BroadcastReceiver có chứa tham số Context trong tham số đầu vào ở các hàm quản lý sự kiện
- ContentProvider gọi hàm getContext để lấy đối tượng Context



## Activity

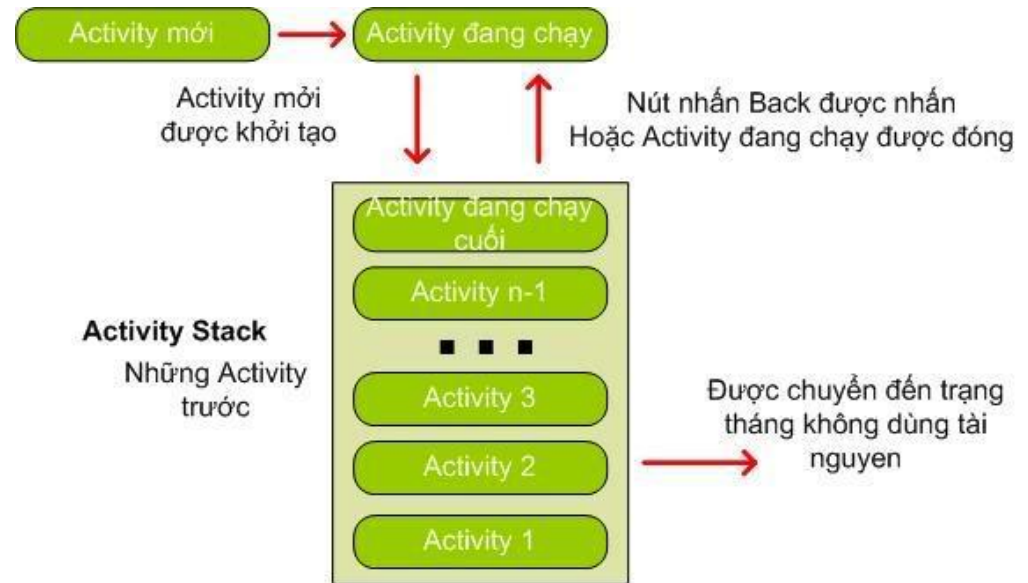
- Khởi tạo Activity bằng cách gọi `startActivity(Intent)`
- Subactivity: Là activity được gọi bởi activity khác.
- Gọi Subactivity sử dụng phương thức `startActivityForResult`
  - Truyền Intent và integer code trong tham số đầu vào
  - Khi subactivity kết thúc, trả lại mã code
  - `startActivityForResult` là phương thức không đồng bộ





# Activity

- Các activity cấu tạo nên Stack
  - Activity mới sẽ xuất hiện ở đầu Stack
  - Thông thường, khi nhấn nút back sẽ quay lại activity trước đó



## Task

- Android nhóm các activity trong một chương trình vào một công việc chung (hàng đợi các activity liên quan đến nhau)
- Người dùng nhấn nút HOME và khởi tạo một chương trình mới
  - Chuyển task hiện tại sang chế độ nền
  - Bắt đầu task mới, đặt activity mặc định của ứng dụng mới ở đầu Stack
- Nếu ứng dụng được quay lại, task cũ (stack cũ) sẽ được khôi phục

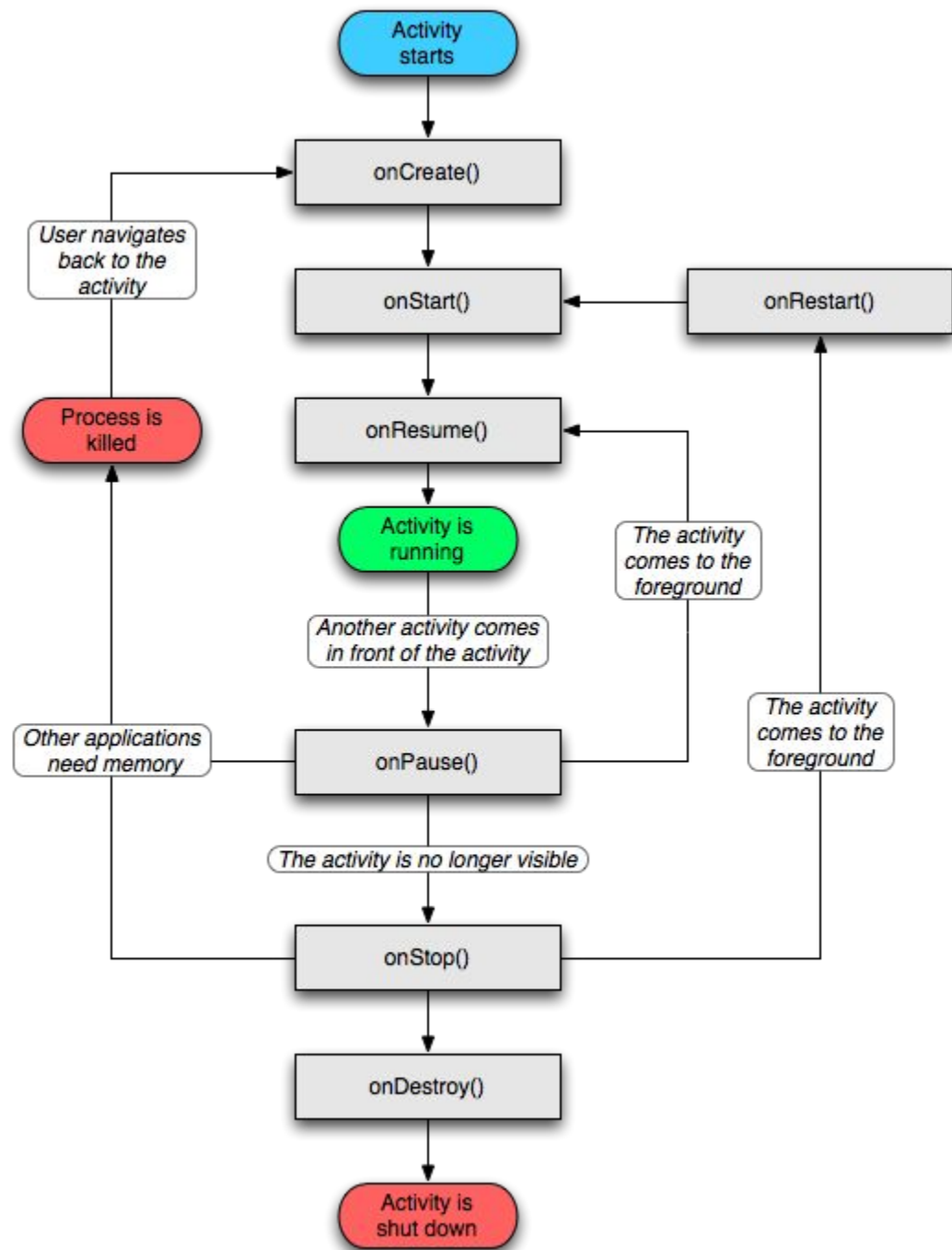


## Vòng đời của Activity

- startActivity đảm bảo Activity được khởi tạo
  - Nếu Activity được khởi tạo, sẽ được đưa lên đầu
  - Activity được quản lý như thế nào?
- Mô hình hướng sự kiện
  - Activity có một số hàm để điều khiển các sự kiện
    - onCreate, onResume, onPause,...
  - Tất cả Activity phải nạp chồng hàm onCreate để thực hiện một việc gì đó
  - Các hàm nạp chồng phải gọi phương thức của superclass



# Vòng đời của Activity



## Vòng đời của Activity

- Ba trạng thái
  - Kích hoạt (active): ở chế độ nền, đang chạy
  - Tạm dừng (pause): vẫn hiển thị nhưng bị che khuất bởi Activity khác
    - Giống active, nhưng có thể bị hủy nếu dung lượng bộ nhớ thấp
  - Dừng (stop): không hiển thị trên màn hình



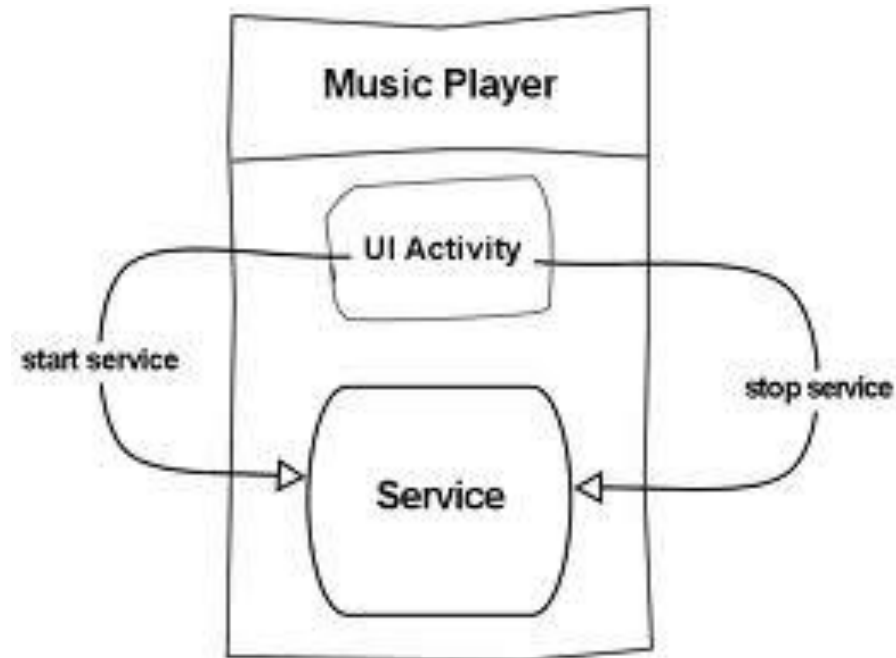
## Vòng đời của Activity

- onCreate()
  - Gọi khi Activity đầu tiên được tạo
  - Chuẩn bị GUI và các bước khởi tạo
  - khác
- onResume()
  - Gọi khi Activity ở trên đầu Stack
  - Cập nhật giá trị GUI
  - Chú ý: được gọi khi Activity đầu tiên được khởi tạo
- onPause()
  - Activity chuẩn bị biến mất
  - Cập nhật các dữ liệu quan trọng, dừng các công việc tốn nhiều tài nguyên

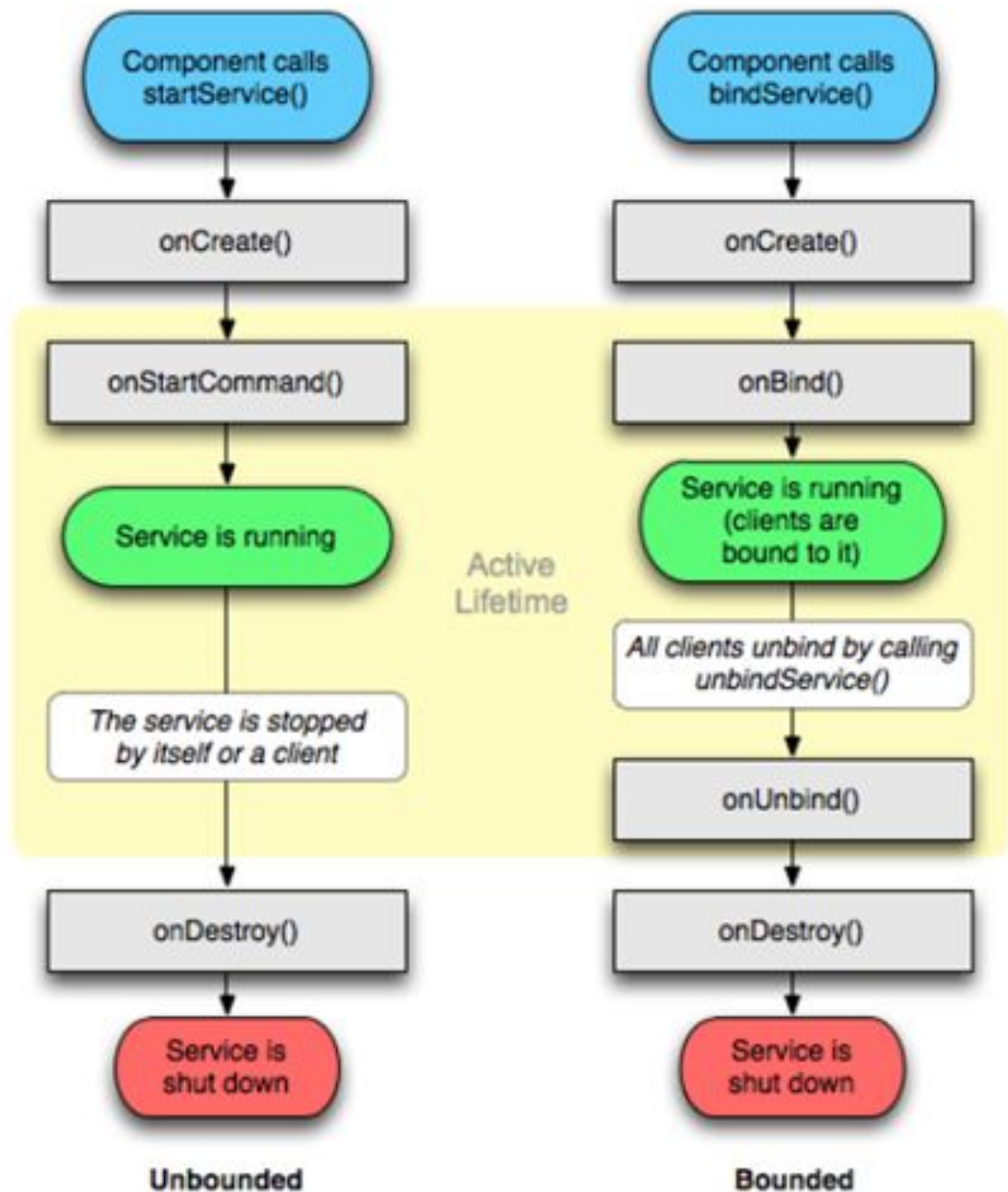


## Vòng đời của Service

- Hai loại:
  - Làm một số công việc nền theo yêu cầu
    - Gọi `startService()`
    - Service có hàm `StartCommand()` hoặc `onStart` để điều khiển
    - Service tiếp tục chạy sau khi lệnh được thi hành
  - Truyền thông
    - Ví dụ: trình nghe nhạc
    - Sử dụng `bindService` để tạo kết nối vững chắc
    - Client nhận đối tượng và gọi Service



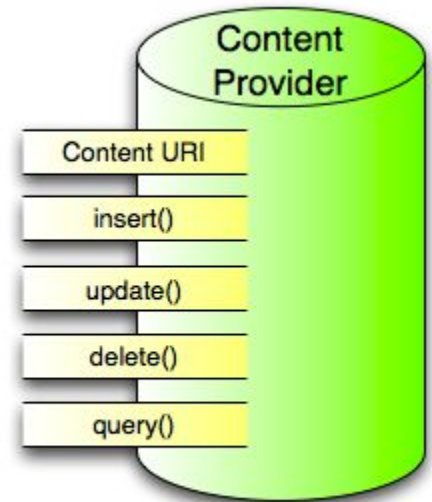
## Vòng đời của Service





## ContentProvider

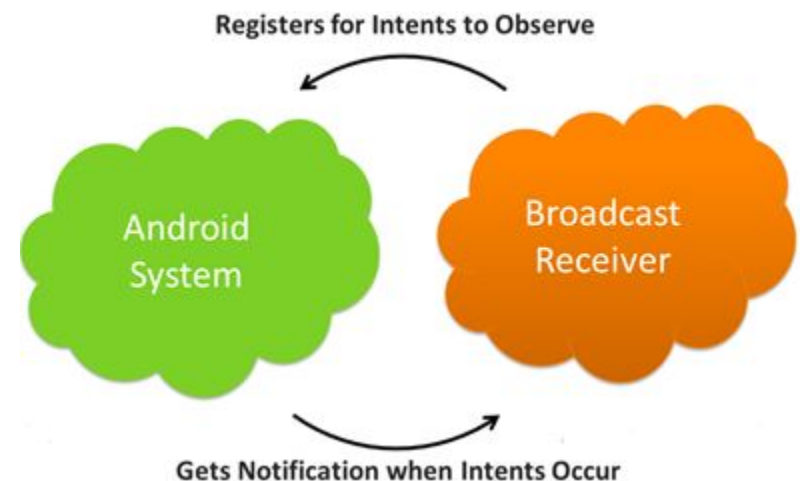
- Truy cập thông qua truy vấn content://URI
  - Có truy vấn, thêm mới, xóa,...
  - Tìm hiểu sâu hơn các bài sau



```
ContentResolver cr=  
Context.getContentResolver();  
cr.query(content://android.provider.  
Contacts.Phones.CONTACT_URI,...)
```

## BroadcastReceiver

- Đánh thức bởi broadcast hệ thống
- Rất đơn giản – chỉ là onReceive handler
  - Nhận context và Intent miêu tả broadcast

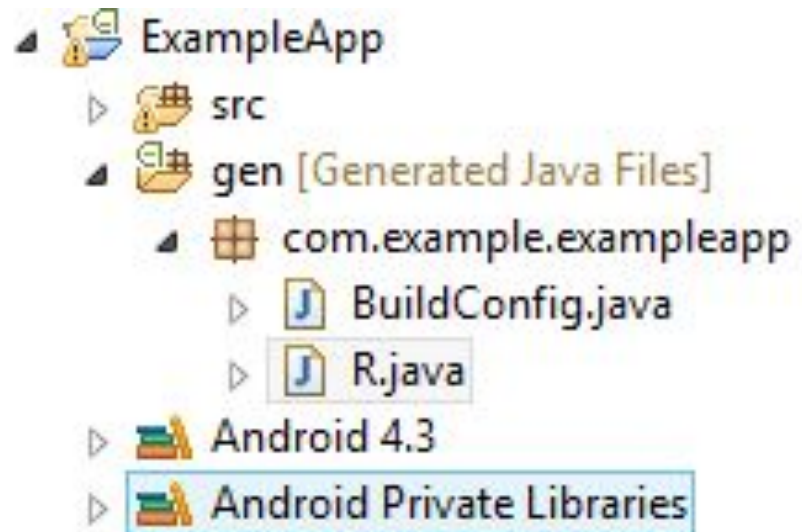


## Tài nguyên của ứng dụng

- Không nằm trong code
  - Ví dụ: String, image
  - Dễ dàng trong việc hỗ trợ các cấu hình thiết bị khác nhau
- Các file chung trong thư mục res/
  - • **drawable/icon.png**: biểu tượng của chương trình trong launcher
  - **layout/activity\_main.xml**: giao diện chính của main Activity
  - **values/strings.xml**: chứa các chuỗi xuất hiện trên UI
  -

## R.java

- Nằm trong thư mục gen, dùng để truy cập đến các tài nguyên trong code
- Ví dụ
  - `R.string.<string_name>`, `R.layout.<layout_name>`
- Truy cập các tài nguyên thông qua lớp R giúp cho Android quyết định tài nguyên nào là phù hợp

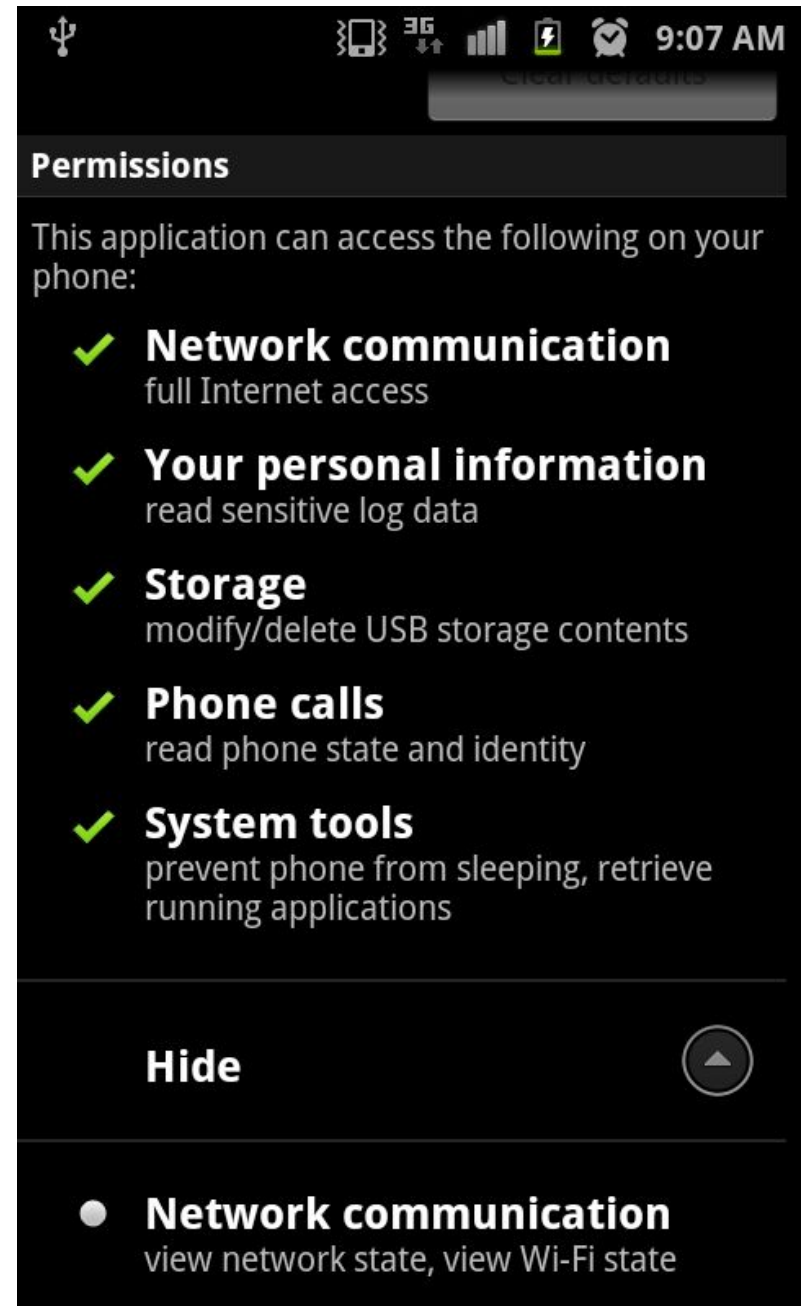


## Thay đổi cấu hình

- Hành vi mặc định của Android: Nếu thay đổi cấu hình, khởi tạo lại Activity
  - Ví dụ: quay màn hình
- Thi thoảng, cách tiếp cận nào tốn nhiều thời gian
  - Điều gì xảy ra nếu Activity tốn nhiều thời gian để tải dữ liệu?
  - Có thể nạp chồng `onConfigurationChanged()` để ngăn cản restart và vẫn điều khiển được các thay đổi

# Android manifest

- File XML
- Metadata về ứng dụng và các thành phần của nó
- Ứng dụng
  - Name, icon, version, version android bắt buộc
  - Quyền mà ứng dụng yêu cầu
  - Đặc trưng mà ứng dụng cần hoặc sử dụng



## Android manifest

- Đối với mỗi thành phần
  - Xác định intent filter do đó Android sẽ biết Intent mà mỗi thành phần có thể điều khiển
- Activity
  - Xác định hành động `android.intent.action.MAIN` và category `android.intent.category.LAUNCHER` cho activity mặc định – hiển thị trên launcher
  - Xác định các hành động khác mà activity có thể điều khiển, ví dụ kiểu của file có thể xem, hoặc URL có thể truy cập
- BroadcastReceiver
  - Xác định sự kiện mà receiver muốn điều khiển

# Android manifest

The screenshot shows the Eclipse IDE with the BlackBerry Java Plug-in. The main editor displays the 'AndroidManifest Application' view. The 'Application Nodes' list shows '.HomeActivity' selected. The 'Add...' button is highlighted with a red box. The 'Intent Filter' button is also highlighted with a red box. The 'Manifest' tab at the bottom shows the 'Application' element selected. The 'Package Explorer' on the left shows the project structure, with 'AndroidManifest.xml' highlighted in a red box. The 'Properties' view on the right shows the 'Create a new element at the top level, in Application' dialog. The 'Intent Filter' button is highlighted with a red box. The 'Manifest' tab at the bottom shows the 'Application' element selected. The 'Package Explorer' on the left shows the project structure, with 'AndroidManifest.xml' highlighted in a red box. The 'Properties' view on the right shows the 'Create a new element at the top level, in Application' dialog. The 'Intent Filter' button is highlighted with a red box.

then finally add  
**action**  
and  
**category** as per selected  
option

Launching DatabaseDemo

3:53 PM  
6/23/2011



## Ứng dụng Android

- Ứng dụng Android là gì?
- Câu trả lời: file apk
  - Tương đương file jar
- Chứa code, tài nguyên, và tất cả các thứ cần cho ứng dụng



# Tổng kết nội dung bài học

- Intent, Content
  - Activity và vòng đời của Activity
  - Service và vòng đời của Service
  - Tài nguyên ứng dụng
- <https://www.youtube.com/@LapTrinhJava>