



BÀI 4:

LẬP TRÌNH ANDROID CƠ BẢN

**INTENT – INTENT FILTER
– CUSTOM LISTVIEW**

- ⊙ Kết thúc bài học này bạn có khả năng
 - ⊙ Hiểu rõ về Intent và Intent Filter
 - ⊙ Custom ListView



Phần I: Intent và Intent Filter

 Intent

 Intent Filter

Phần II: Custom ListView

 Các bước custom ListView

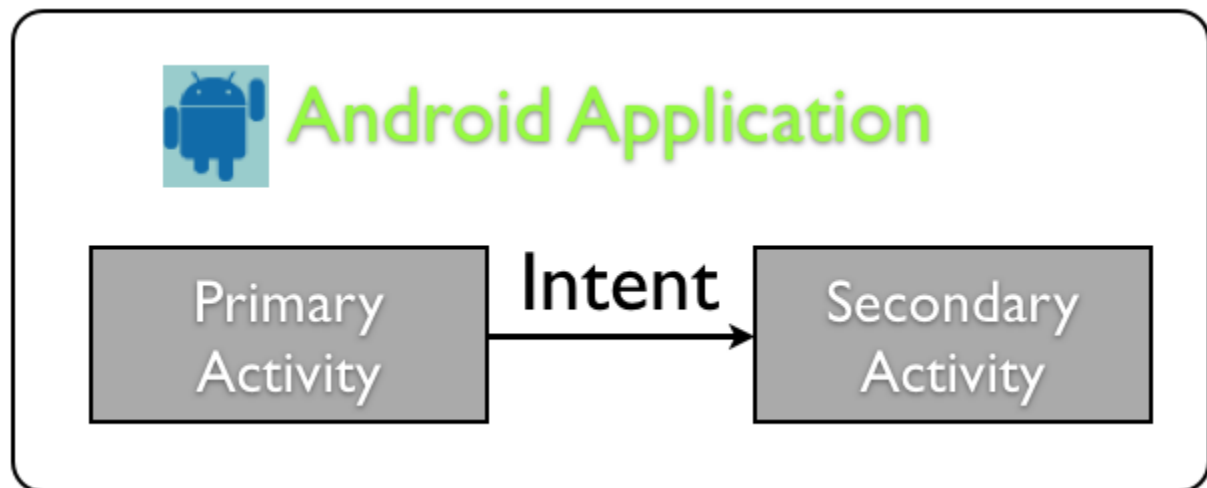


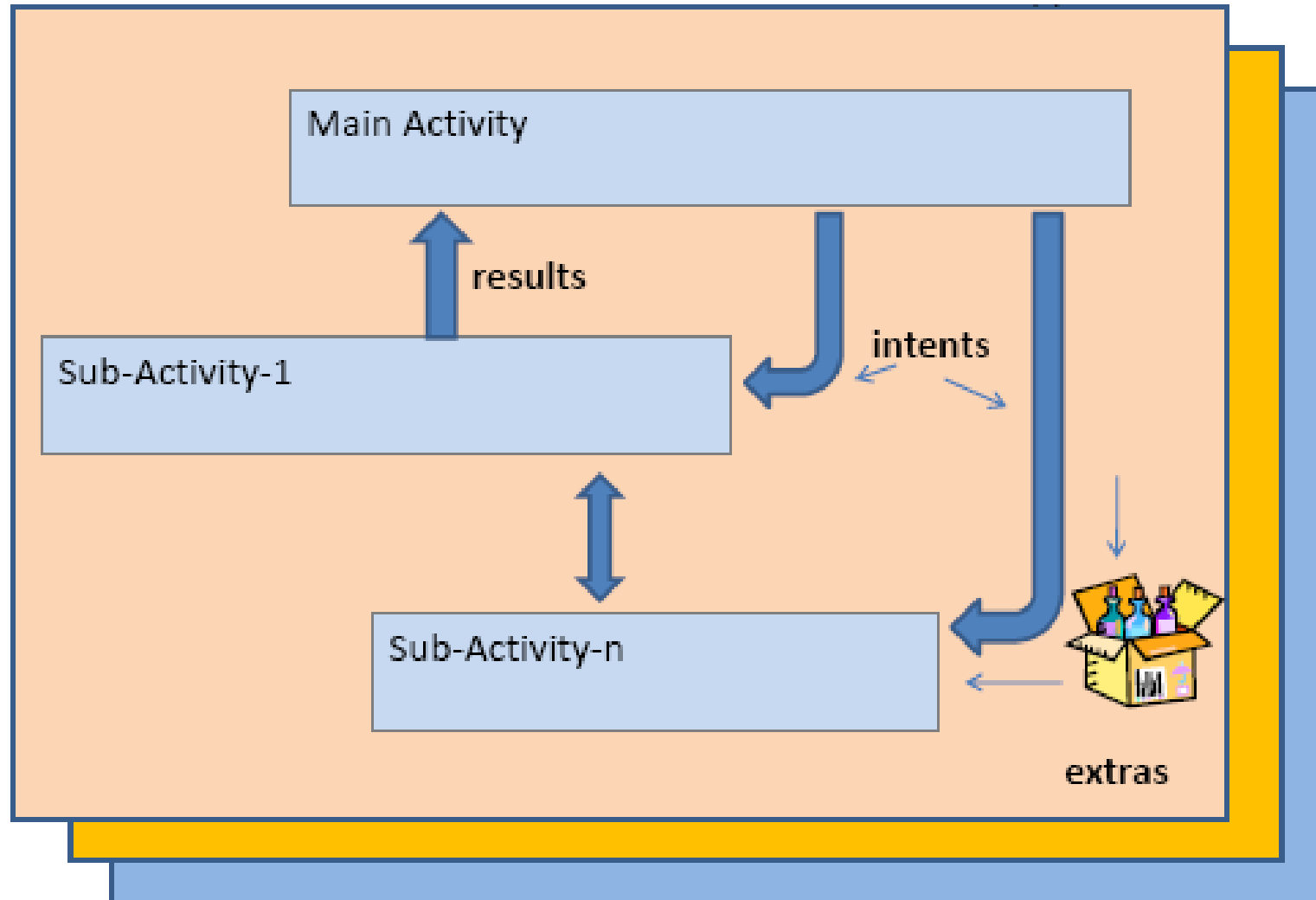


BÀI 4: INTENT – INTENT FILTER – CUSTOM LISTVIEW

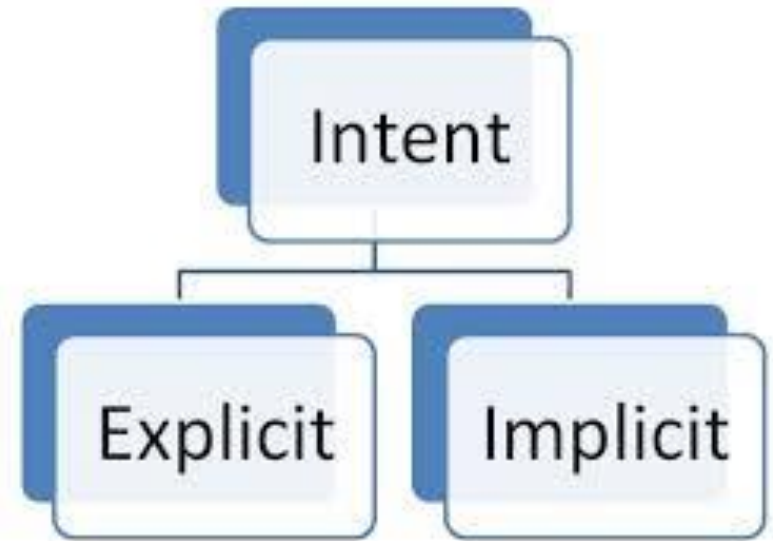
PHẦN 1: INTENT VÀ INTENT FILTER

- Các thành phần chính của ứng dụng (activity, service, broadcast receiver) được kích hoạt thông qua thông điệp gọi là Intent
- Thông điệp Intent ràng buộc giữa các thành phần trong cùng ứng dụng hoặc các ứng dụng khác nhau
- Di chuyển từ Activity này đến Activity khác sử dụng Intent





- Android hỗ trợ hai loại intent là **explicit intent** và **implicit intent**



Explicit Intent

- Explicit intent: được khai báo một cách tường minh thành phần sẽ nhận và xử lý Intent bằng cách thiết lập giá trị phù hợp
- Explicit intent thường được sử dụng để khởi tạo các activity trong cùng 1 ứng dụng

```
txtSoThuNhat = (EditText)findViewById(id.txtsothunhat);  
txtSoThuHai = (EditText)findViewById(id.txtsothuhai);  
//Tạo intent để mở ResultActivity  
Intent intent = new Intent(this, ResultActivity.class);  
//Khai báo Bundle  
Bundle bundle = new Bundle();  
int a = Integer.parseInt(txtSoThuNhat.getText().toString());  
int b = Integer.parseInt(txtSoThuHai.getText().toString());  
//Truyền dữ liệu vào Bundle  
bundle.putInt("sothunhat", a);  
bundle.putInt("sothuhai", b);  
//Đưa Bundle vào Intent  
intent.putExtra("mypkg", bundle);  
//Mở Activity  
startActivity(intent);
```


Implicit Intent

- Là loại intent không cần chỉ rõ thành phần xử lý mà chỉ cần cung cấp đủ các thông tin cần thiết để hệ thống xác định xem nên dùng các thành phần có sẵn nào để chạy tốt nhất
- Ví dụ: khi ứng dụng Android mở một trang web

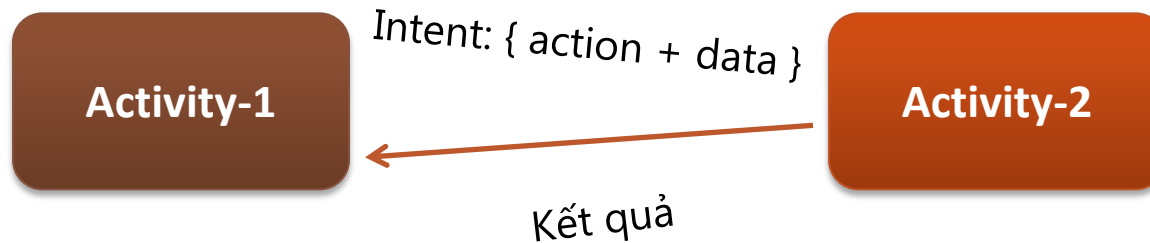
```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://google.net"));
startActivity(intent);
```

- Intent được gọi sử dụng các cách sau:
 - startActivity(intent) hoặc startActivityForResult(): khởi tạo một Activity hoặc khởi tạo Activity và lấy kết quả trả về
 - sendBroadcast(intent): gửi một Intent tới thành phần BroadcastReceiver quan tâm đến
 - startService(intent) hoặc bindService(intent,...): thiết lập kết nối giữa thành phần được gọi và dịch vụ đích

Đối tượng Intent

- Đối tượng Intent là một gói các thông tin
- Chứa thông tin hành động sẽ diễn ra và dữ liệu liên quan đến hành động đó
- Tham số chính của Intent bao gồm:
 - Action: kiểu chuỗi chứa thông tin hành động sẽ diễn ra
 - Data: dữ liệu chính cần thao tác, ví dụ như số điện thoại được gọi (mô tả dưới dạng URI)
 - Extra: cặp giá trị key-value chứa thông tin bổ sung được truyền tới thành phần điều khiển intent
- Ngoài ra còn một số tham số khác như category, type, component (category dùng để đưa thông tin bổ sung cho hành động sẽ được thi hành)

Tham số chính của Intent



```
Intent myActivity= new Intent(action, data);  
startActivity(myActivity);
```

Activity sẵn có hoặc do người dùng tự tạo

Dữ liệu chính (dưới dạng URI)
tel://
http://
sendto://

Tham số Action

- ACTION_CALL
- ACTION_EDIT
- ACTION_MAIN
- ACTION_SYNC
- ACTION_BATTERY_LOW
- ACTION_HEADSET_PLUG
- ACTION_SCREEN_ON
- ACTION_TIMEZONE_CHANGED

Tham số Category

- CATEGORY_BROWSABLE
- CATEGORY_GADGET
- CATEGORY_HOME
- CATEGORY_LAUNCHER
- CATEGORY_PREFERENCE

Ví dụ về cặp giá trị Action/Data

- **ACTION_CALL** <tel:123>: thực hiện gọi điện thoại
- **ACTION_VIEW** <http://www.google.com>: hiển thị trang chủ của google trên trình duyệt
- **ACTION_EDIT** **content://contacts/people/2**: sửa đổi thông tin liên hệ có định danh là 2
- **ACTION_VIEW** **content://contacts/people/**: hiển thị danh sách người cho phép người dùng lựa chọn

URI

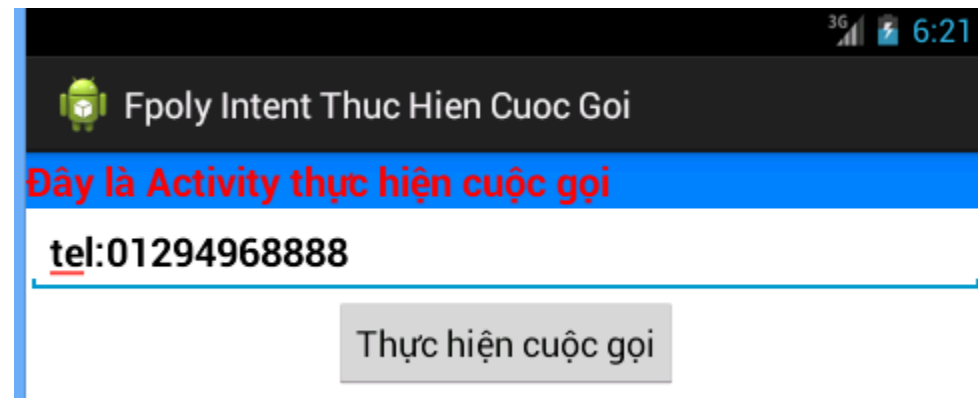
- URI (Uniform Resource Identifier) là một dãy các ký tự dùng để
 - Mở một trang web trong trình duyệt, bắt đầu với "http:"
 - Chỉ tới các file tài nguyên có sẵn trong máy (ví dụ 1 file video, danh sách contacts,... bắt đầu với "content:"

`content://com.example.transportationprovider/trains/122`

The diagram shows the URI `content://com.example.transportationprovider/trains/122` with four brackets underneath it, labeled A, B, C, and D. Bracket A is under `content:`, bracket B is under `//com.example.transportationprovider`, bracket C is under `/trains`, and bracket D is under `/122`.

Ví dụ

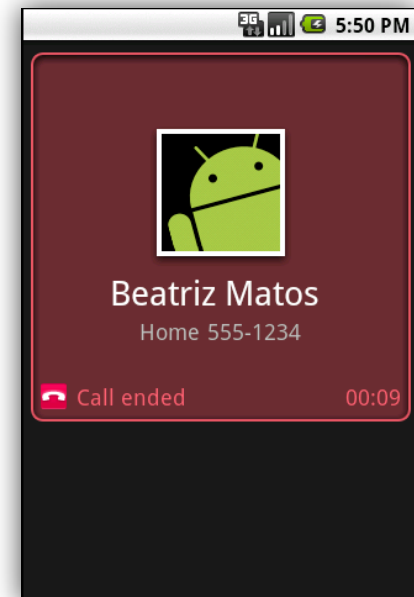
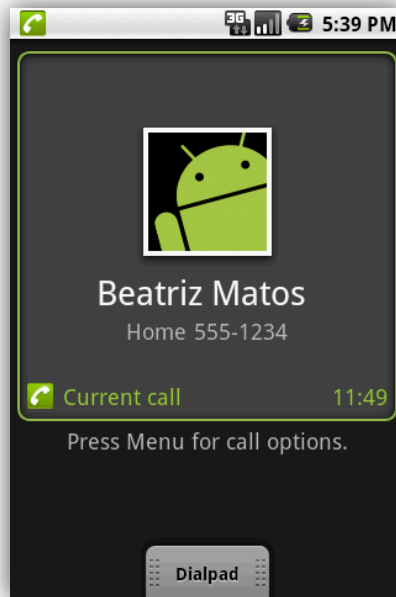
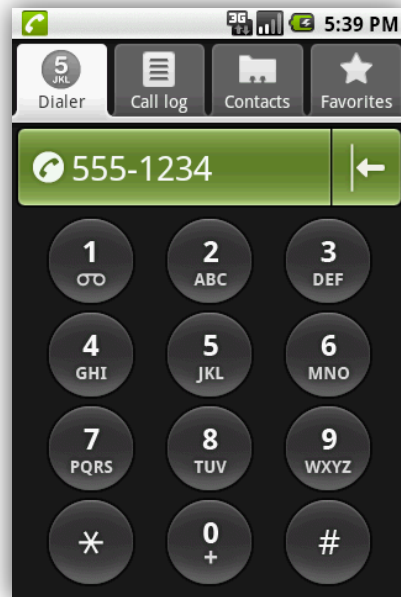
- Activity1 hiển thị giao diện cho phép nhập số điện thoại và yêu cầu Activity2 (sẵn có của hệ thống) thực hiện cuộc gọi
- Chú ý để thực hiện cuộc gọi phải bổ sung quyền **android.permission.CALL_PHONE** vào trong Android Manifest



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="fpoly.android.com"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />
    <uses-permission android:name="android.permission.CALL_PHONE"/>
```

Ví dụ



```

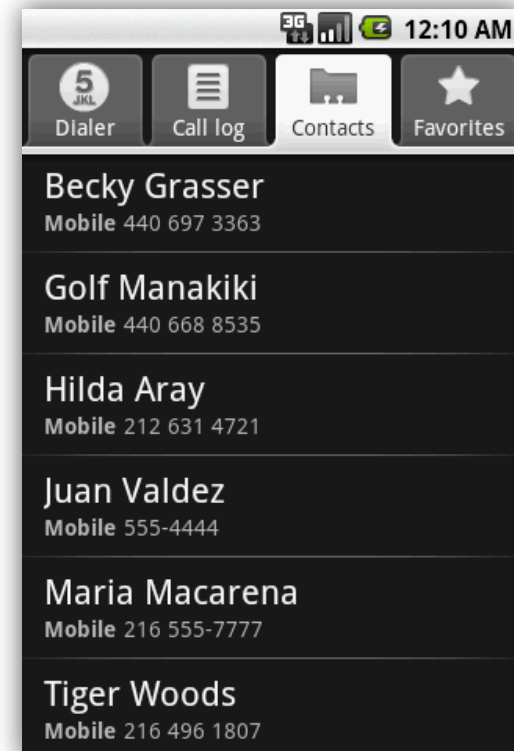
11 public class MainActivity extends Activity {
12     EditText txtPhoneNumber;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17     }
18
19     @Override
20     public boolean onCreateOptionsMenu(Menu menu) {
21         // Inflate the menu; this adds items to the action bar if it is present.
22         getMenuInflater().inflate(R.menu.main, menu);
23         return true;
24     }
25     public void thucHienCuocGoi(View v)
26     {
27         txtPhoneNumber = (EditText)findViewById(R.id.txtNumber);
28         String number = txtPhoneNumber.getText().toString();
29         Intent activityPhoneCall = new Intent(Intent.ACTION_CALL, Uri.parse(number));
30         startActivity(activityPhoneCall);
31     }
32 }

```

Ví dụ

- Hiển thị tất cả contact trên thiết bị để người dùng lựa chọn

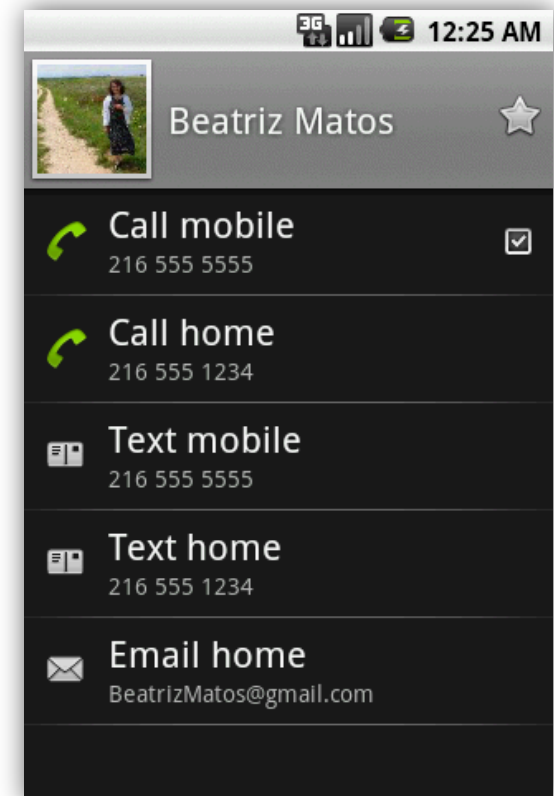
```
String myData= "content://contacts/people/";  
Intent myActivity2 = new Intent(Intent.ACTION_VIEW,  
Uri.parse(myData));  
startActivity(myActivity2);
```



Ví dụ

- Hiển thị thông tin của một contact cụ thể

```
String myData= "content://contacts/people/2";  
Intent myActivity2 = new Intent(Intent.ACTION_VIEW,  
Uri.parse(myData));  
startActivity(myActivity2);
```



Ví dụ

- Sửa thông tin một contact cụ thể

```
String myData= "content://contacts/people/1";  
Intent myActivity2 = new Intent(Intent.ACTION_EDIT,  
Uri.parse(myData));  
startActivity(myActivity2);
```

The screenshot shows an Android contact editing interface. At the top, there's a blue header with a checkmark and the word 'DONE'. Below it, a grey bar indicates 'Phone-only, unsynced contact'. The contact's name 'Nguyen Van An' is in a text field, followed by a dropdown arrow and a profile picture icon. Below the name is an 'Add organization' link. The 'PHONE' section shows '(098) 344-4456' with a 'MOBILE' label and a close button. Below that is an 'Add new' link. The 'EMAIL' section shows 'nguyenvanan@gmail.com' with a 'HOME' label and a close button. Below that is another 'Add new' link. The 'ADDRESS' section shows '73 Le Hong Phong' with a 'HOME' label and a close button. At the bottom is an 'Add new' link.

Bundle

- Sử dụng đối tượng Bundle để truyền dữ liệu
- Bundle có 2 loại phương thức putA dùng để cho bên gửi đi, ứng với kiểu dữ liệu nào thì put đúng kiểu đó (A là kiểu dữ liệu).
- Ví dụ: putInt và putDouble là tương ứng với 2 loại dữ liệu khác nhau
- Ở bên nhận, Bundle dùng phương thức getA (A là kiểu dữ liệu) để lấy dữ liệu theo key bên gửi
- Key phải giống nhau cho cả 2 bên nhận và gửi

Ví dụ về Bundle

1. Gửi data vào Bundle

```
//Tạo intent để mở ResultActivity
Intent intent = new Intent(this, ResultActivity.class);
//Khai báo Bundle
Bundle bundle = new Bundle();
int a = Integer.parseInt(txtSoThuNhat.getText().toString());
int b = Integer.parseInt(txtSoThuHai.getText().toString());
//Truyền dữ liệu vào Bundle
bundle.putInt("sothunhat", a);
bundle.putInt("sothuhai", b);
//Đưa Bundle vào Intent
intent.putExtra("mypkg", bundle);
//Mở Activity
startActivity(intent);
```


2. Lấy data từ Bundle

```
//Lấy intent đã gọi Activity này
Intent myIntent = getIntent();
//Lấy bundle dựa vào tên package
Bundle myBundle = myIntent.getBundleExtra("mypkg");
//Lấy thông tin của số thứ nhất và số thứ hai
int a = myBundle.getInt("sothunhat");
int b = myBundle.getInt("sothuhai");
```



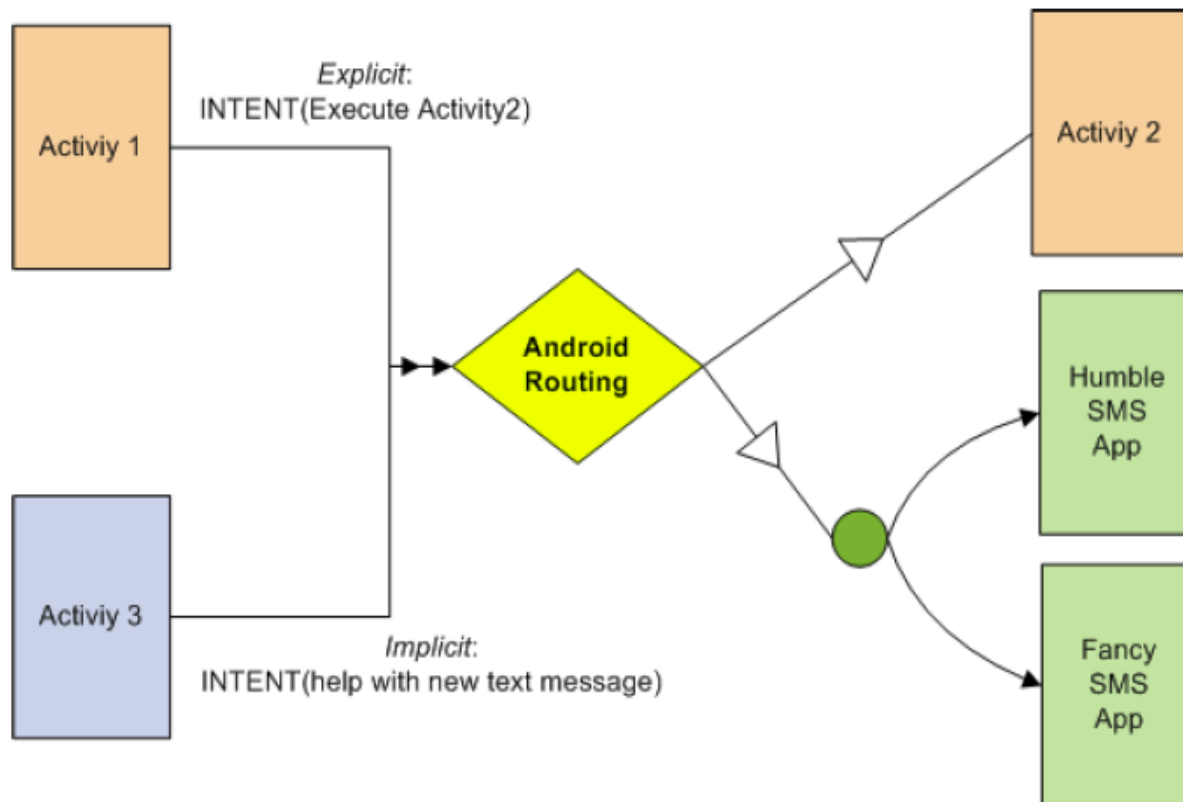
DEMO

- Truyền dữ liệu qua Bundle



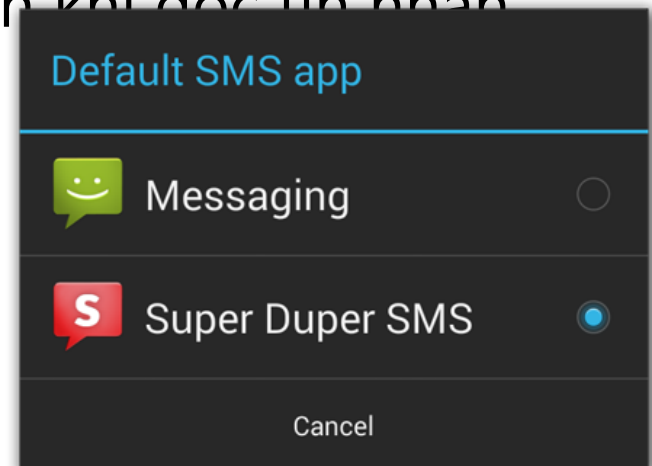
- Để thông báo cho hệ thống implicit intent nào được sử dụng, activity, service và broadcast receiver có thể có một hoặc nhiều bộ lọc intent (intent filter)

Intent Resolution



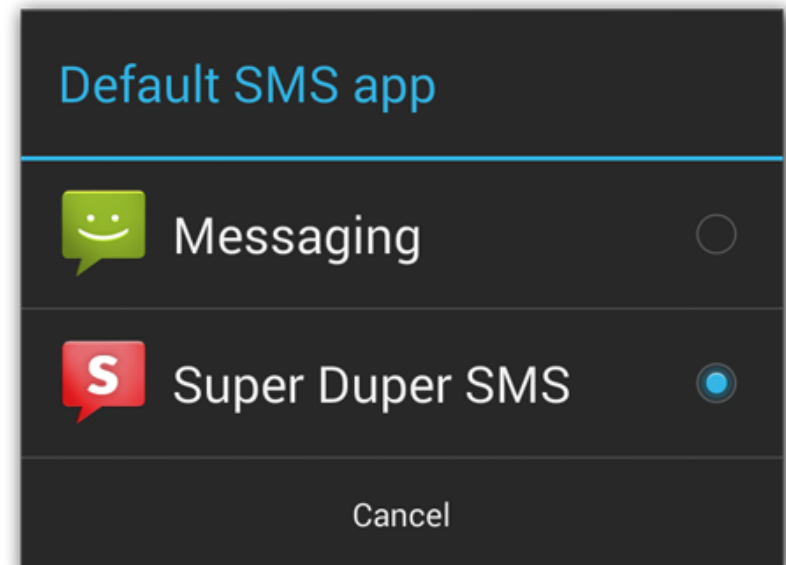
Ví dụ về Intent Filter

- Giả sử người dùng cài ứng dụng SMS "Super Duper SMS" để thay thế cho ứng dụng Messaging mặc định của máy
- Với sự ra đời của Implicit Intent, Android sẽ "hỏi" người dùng câu hỏi sau "Bạn có một tin nhắn mới. Bạn có hai ứng dụng nhắn tin "Super Duper SMS" và ứng dụng Messaging mặc định. Bạn chọn phần mềm nào, bạn có muốn chọn phần mềm này là phần mềm mặc định khi đọc tin nhắn không?"



Ví dụ về Intent Filter

- Một activity được gọi khi thỏa mãn các điều kiện sau:
 - Hỗ trợ action cho trước
 - Hỗ trợ MIME type đã được miêu tả (nếu cung cấp)
 - Hỗ trợ tất cả category được nêu trong Intent



Ví dụ Intent Filter

- Phần tử <data> dưới đây sẽ thông báo cho Android biết rằng component có thể nhận dữ liệu video từ trên mạng và có thể hiển thị

```
<data android:scheme="http" android:type="video/*" />
```

- Android sẽ chọn một trình duyệt khi người dùng click vào một link web
- Đầu tiên, android sẽ hiển thị dữ liệu (nếu link trên trang web). Nếu không thể hiển thị dữ liệu, nó sẽ đẩy implicit intent cùng với scheme và data type và thử khởi tạo một activity đáp ứng được công việc này
- Nếu không có Activity đáp ứng được, trình download manager sẽ tải dữ liệu về

Ví dụ Intent Filter

- Ví dụ dưới đây cho Android biết component có thể nhận image data từ content provider và hiển thị:

```
<data android:mimeType="image/*" />
```

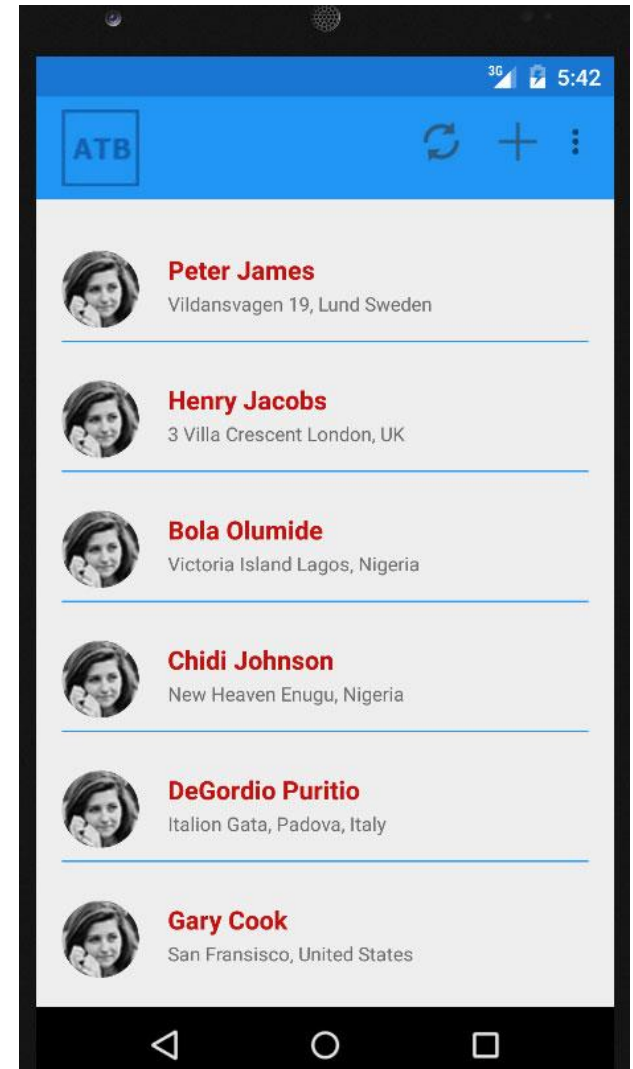
- Đa số dữ liệu được phân phối bởi content provider, filter chủ yếu xác định kiểu dữ liệu



BÀI 4: INTENT – INTENT FILTER – CUSTOM LISTVIEW PHẦN II: CUSTOM LISTVIEW

Các bước thực hiện:

1. Thiết kế layout custom cho mỗi dòng.
2. Viết Class quản lý các thuộc tính cần show trên các dòng ListView
3. Viết Adapter để getView phù hợp vào Layout đã custom.
4. Thực hiện các bước còn lại giống ListView





Phần I: Intent và Intent Filter

 Intent

 Intent Filter

Phần II: Custom ListView

 Các bước custom ListView





Cảm ơn