



LẬP TRÌNH THIẾT BỊ DI ĐỘNG

Mai Cường Thọ

Bài 5

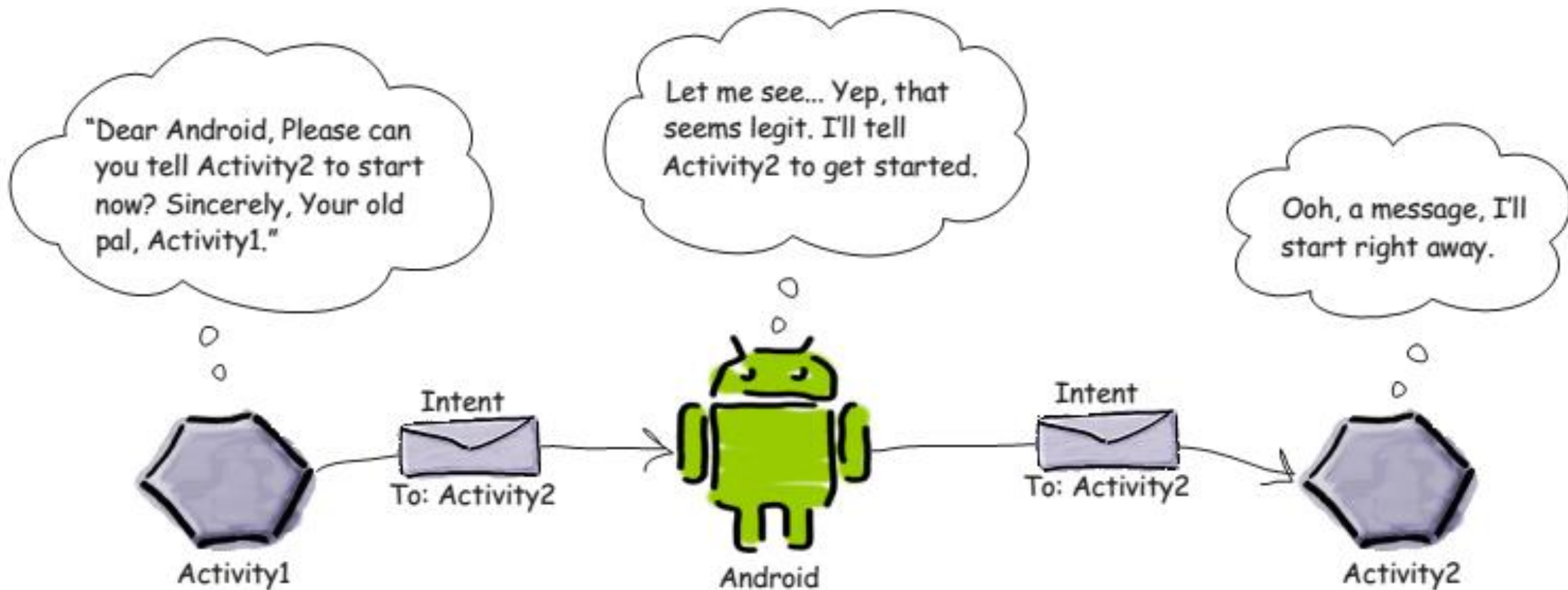
Liên kết các activity sử dụng intents

Nội dung



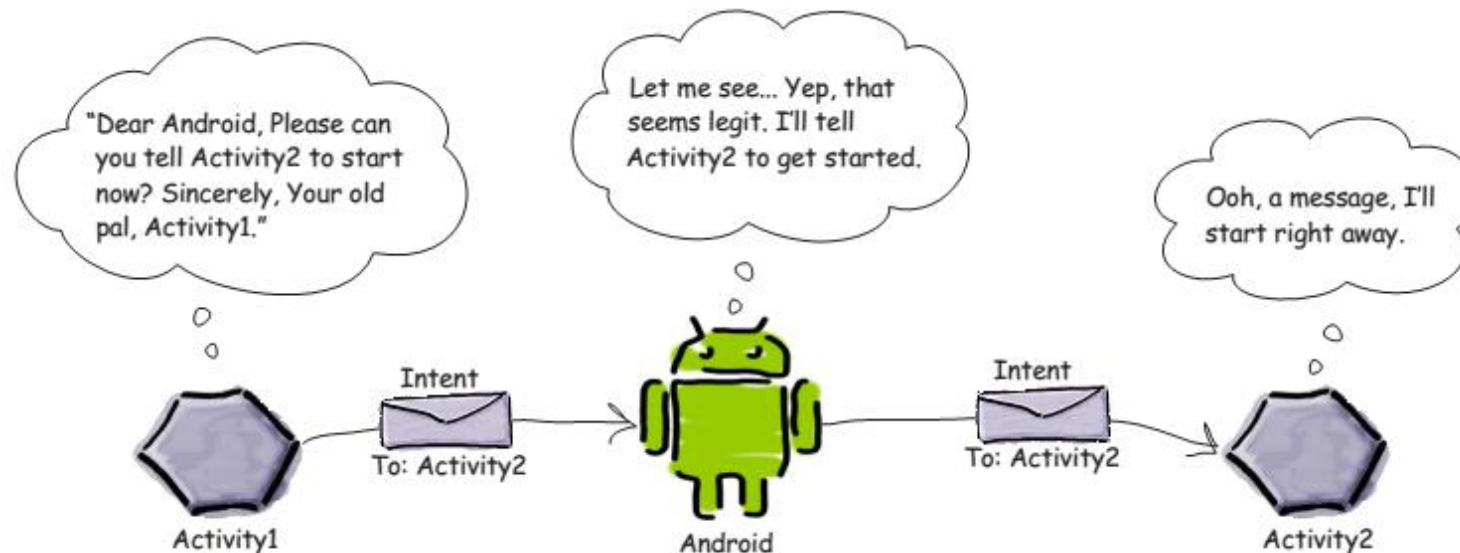
1. Giới thiệu về **intent**
2. Sử dụng intent để trao đổi dữ liệu
3. Intent filter
4. Intent tường minh vs ngầm định
5. Các thành phần của intent
 - Action & Data
 - Category
 - Type
 - Component
 - Extras

1. Giới thiệu về intent



Intent & Intent Service

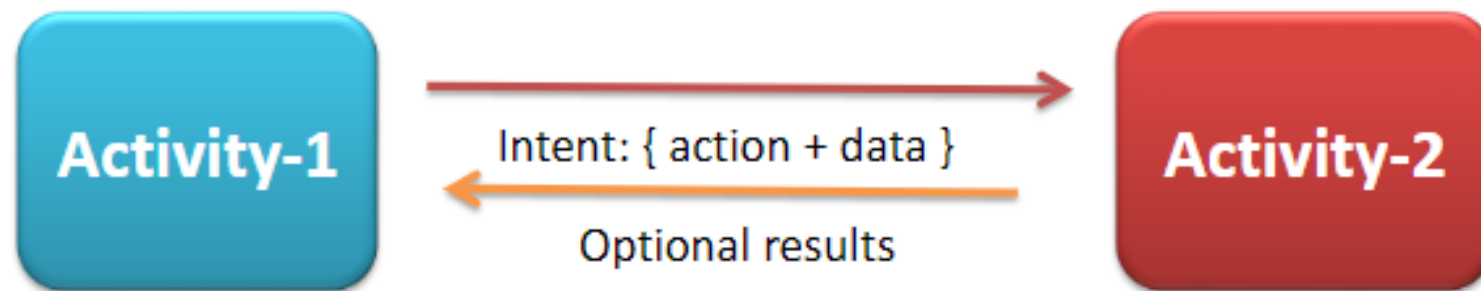
- **Intent**: giúp giao tiếp giữa các thành phần trong Android OS (activity, service, provider, receiver)
- **Intent** giống một lá thư: các thông tin cần thiết được đóng gói bên trong một intent (địa chỉ + nội dung)
- **Intent service**: dịch vụ hệ thống, vai trò như người đưa thư: chuyển intent tới thành phần nhận phù hợp nhất



Parts of an Intent



- The two main components of an Intent are:
- **Action:** The built-in action to be performed, such as **ACTION_VIEW**, **ACTION_EDIT**, **ACTION_CALL**, **ACTION_SENDTO**,... or a user-created-activity
- **Data:** Basic argument needed by the intent to work. For instance: a phone number to be called , a picture to be shown, a message to be sent, etc.



Initiating an Intent



Typically an intent is called as follows:

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

Primary data (as an URI)

tel://
http://
sendto://

Built-in or
user-created activity

Examples of **action**/data pairs:



- | | |
|--------------------|---|
| ACTION_DIAL | <i>tel://5551234</i> or <i>tel:5551234</i> Display the phone dialer with the given number filled in. |
| ACTION_VIEW | <i>http://www.google.com</i> Show Google page in a browser view. |
| ACTION_EDIT | <i>content://contacts/people/2</i> Edit information about the contact person whose identifier is "2". |
| ACTION_VIEW | <i>content://contacts/people/2</i> Used to start an activity to display contact person whose identifier is "2". |
| ACTION_VIEW | <i>content://contacts/people/</i> Display a list of people, which the user can browse through. Selecting a person to view would result in a new intent. |

Common Built-in Android Actions



List of common actions that Intents can use for launching built-in activities
[usually through *startActivity(Intent)*]

ACTION_MAIN

ACTION_VIEW

ACTION_ATTACH_DATA

ACTION_EDIT

ACTION_PICK

ACTION_CHOOSER

ACTION_GET_CONTENT

ACTION_DIAL

ACTION_CALL

ACTION_SEND

ACTION_SENDTO

ACTION_ANSWER

ACTION_INSERT

ACTION_DELETE

ACTION_RUN

ACTION_SYNC

ACTION_PICK_ACTIVITY

ACTION_SEARCH

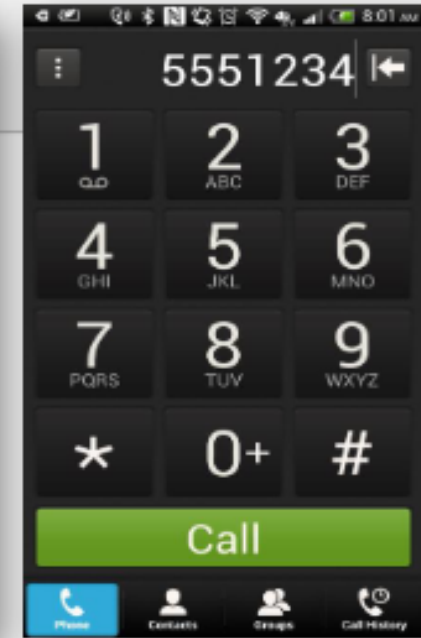
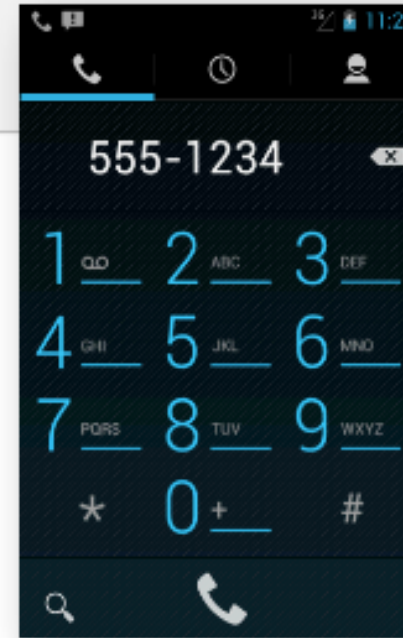
ACTION_WEB_SEARCH

ACTION_FACTORY_TEST

Example1A: ACTION_DIAL

- **ACTION_DIAL** Display the phone dialer with the given number filled in.

```
String myPhoneNumberUri = "tel:555-1234";  
  
Intent myActivity2 = new Intent(Intent.ACTION_DIAL,  
                                Uri.parse(myPhoneNumberUri));  
startActivity(myActivity2);
```



Images captured
from emulator and
device respectively

Example1B: ACTION_CALL

- Placing an immediate phone call

```
String myData = "tel:555-1234";

Intent myActivity2 = new Intent(
    Intent.ACTION_CALL,
    Uri.parse(myData) );

startActivity(myActivity2);
```

Needs Permission: <uses-permission
android:name="android.permission.CALL_PHONE" />



Trao đổi intent giữa các activity



PHÍA ACTIVITY GỬI

- Dữ liệu được đóng gói vào intent
 - Lựa chọn dữ liệu đưa vào intent phù hợp
 - Có chuẩn trao đổi dữ liệu chung để dễ xử lý
- Sau đó gọi activity phù hợp để xử lý
 - `startActivity(myIntent);`
- Nếu muốn nhận kết quả trả về thì dùng
 - `startActivityForResult(myIntent, CODE);`
 - Trong đó `CODE` là một số nguyên, dùng để phân biệt kết quả trả về giữa các activity khác nhau
 - Xử lý bằng cách viết lại hàm `onActivityResult`

Trao đổi intent giữa các activity ..



PHÍA ACTIVITY NHẬN

- Lấy intent gửi cho mình bằng `getIntent()`
- Nếu cần **trả về** kết quả nào đó cho activity trước
 - Thông báo **thất bại**: `setResult(RESULT_CANCELED);`
 - Thông báo **thành công**: `setResult(RESULT_OK, x);`
 - Với x là intent mà sẽ trả ngược lại cho activity gọi
 - Cần đưa dữ liệu vào x trước khi `setResult`
 - Việc đưa dữ liệu sử dụng các hàm `putExtra`
 - Chú ý trường hợp dữ liệu phức tạp: `putSerializable`
 - Có thể sử dụng: lớp `Bundle`

```
public void btnNhap(View v) {  
    Intent i = new Intent(this, NhapSoActivity.class);  
    startActivityForResult(i, 1001);  
}
```

Mở activity NhapSoActivity,
đợi dữ liệu trả về qua mã
số 1001

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == 1001) {  
        if (resultCode == RESULT_OK) {  
            TextView t1 = (TextView) findViewById(R.id.textView1);  
            TextView t2 = (TextView) findViewById(R.id.textView2);  
            TextView t3 = (TextView) findViewById(R.id.textView3);  
            String a = data.getStringExtra("SoA");  
            String b = data.getStringExtra("SoB");  
            t1.setText("A = " + a);  
            t2.setText("B = " + b);  
            t3.setText("Tổng = " + (Integer.parseInt(a) + Integer.parseInt(b)));  
            Toast.makeText(this, "Trả về thành công", Toast.LENGTH_SHORT).show();  
        }  
        else  
            Toast.makeText(this, "Trả về thất bại", Toast.LENGTH_SHORT).show();  
    }  
    else  
        super.onActivityResult(requestCode, resultCode, data);  
}
```

Xử lý dữ liệu trả
về
onActivityResult:
được tự động gọi
khi activity nhập
liệu kết thúc


```

public class NhapSoActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nhap_so);
    }

    public void btnCancel(View v) {
        setResult(RESULT_CANCELED);
        finish();
    }

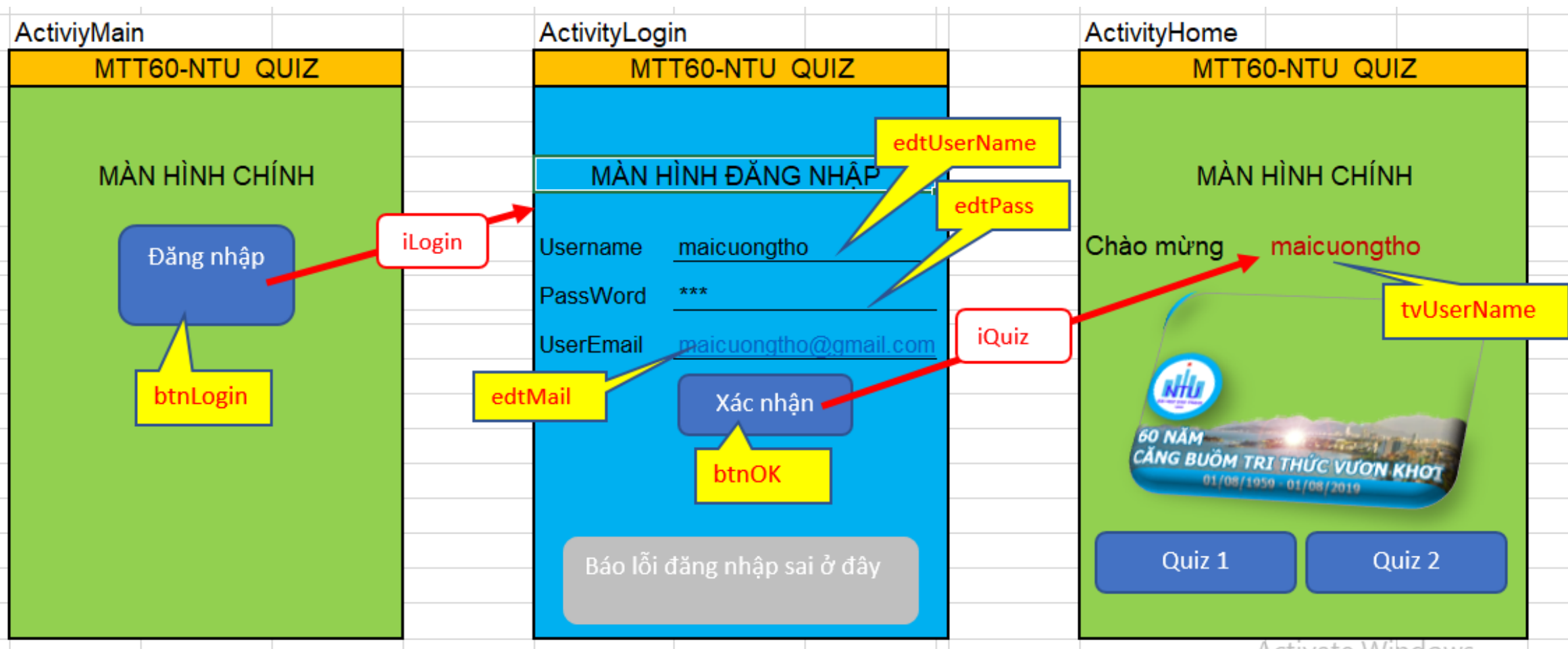
    public void btnOK(View v) {
        Intent i = new Intent();
        EditText t1 = (EditText) findViewById(R.id.editText1);
        EditText t2 = (EditText) findViewById(R.id.editText2);
        i.putExtra("SoA", t1.getText().toString());
        i.putExtra("SoB", t2.getText().toString());
        setResult(RESULT_OK, i);
        finish();
    }
}

```

Xử lý ở phía Activity nhập liệu

Dùng **setResult** để
thiết lập dữ liệu trả về
cho activity đã gọi nó

Thực hành



Intent filter

Intent filter (bộ lọc intent)



- **Activity, Service** và **Broadcast** receiver sử dụng **intent filter** để thông báo cho hệ thống biết các dạng intent mà nó có thể xử lý
- Phân giải intent (intent resolution): khi nhận được một intent, hệ thống tiến hành chọn activity phù hợp nhất với intent đó theo ưu tiên sau
 - Action trong intent
 - Chuỗi tham số (URI trong phần data)
 - Category của intent

Intent filter (bộ lọc intent)..

- Đoạn mã XML dưới đây đăng kí với hệ thống một activity “lắng nghe” việc người dùng muốn thực hiện cuộc gọi bằng cách chỉ định thuộc tính **action** trong **intent-filter**

```
<activity android:name=".Dialer"
          android:label="@string/app_name">
  <intent-filter android:priority="100" >
    <action android:name="android.intent.action.DIAL" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="tel"/>
  </intent-filter>
</activity>
```



Intent tương minh/ ngầm định

Intent tường minh (explicit)



- Sử dụng thành phần component để chỉ định rõ đối tượng sẽ thực thi
- Sử dụng phương thức:
 - `setComponent(ComponentName)`
 - `setClass(Context, Class)`
 - `setClassName(Context, String)`
 - `setClassName(string, string)`
- Chỉ được dùng để gọi các activity trong cùng một ứng dụng
- Dữ liệu trao đổi nên chuyển vào phần extras

Intent tường minh (explicit) ..



```
Intent intentABC = new Intent();  
intent.setClassName("ten_package", "ten_class");  
// chuẩn bị dữ liệu trước khi gửi  
...  
startActivity(intentABC);
```

Hoặc

```
Intent intentABC = new Intent(this, Activity2.class);  
// chuẩn bị dữ liệu trước khi gửi  
...  
startActivity(intentABC);
```

Intent ngầm định (implicit)



- Dùng các thành phần action, category,...
- Hệ thống tự động xác định đối tượng phù hợp nhất để đáp ứng với Intent đó (theo nguyên tắc “phân giải intent” đã trình bày)
- Dùng để giao tiếp với các dịch vụ hệ thống hoặc dịch vụ do bên thứ ba cung cấp:
 - Gọi **activity**: startActivity / startActivityForResult
 - Gọi **service**: startService / bindService
 - Gửi **broadcast**: sendBroadcast
sendOrderedBroadcast
setStickyBroadcast

Intent ngầm định (implicit) ..

- Một số trường hợp sử dụng implicit intent

| Định dạng | Action | Mô tả |
|--|-------------------|--|
| tel:phone_number | ACTION_VIEW | Mở Dial form (chưa gọi) |
| tel:phone_number | ACTION_CALL | Thực hiện gọi tới số phone |
| http://web_address https://web_address | ACTION_VIEW | Mở trình duyệt web với địa chỉ được cấp |
| "some_words" (string) http://web_address https://web_address | ACTION_WEB_SEARCH | Thực hiện search |
| sms:// | ACTION_SENDTO | Gửi tin nhắn |
| geo:latitude,longitude geo:latitude,longitude?z=zoom geo:0,0?q=my+street+address geo:0,0?q=business+near+city | ACTION_VIEW | Mở ứng dụng Maps và chỉ tới vị trí được xác định |

Các action được định nghĩa sẵn

Built-in Standard Actions

[ACTION MAIN](#)

[ACTION VIEW](#)

[ACTION ATTACH DATA](#)

[ACTION EDIT](#)

[ACTION PICK](#)

[ACTION CHOOSER](#)

[ACTION GET CONTENT](#)

[ACTION DIAL](#)

[ACTION CALL](#)

[ACTION SEND](#)

[ACTION ANSWER](#)

[ACTION INSERT](#)

[ACTION DELETE](#)

[ACTION RUN](#)

[ACTION SYNC](#)

[ACTION PICK ACTIVITY](#)

[ACTION SEARCH](#)

[ACTION WEB SEARCH](#)

[ACTION FACTORY TEST](#)

[ACTION SENDTO](#)

Built-in Standard Broadcast Actions

[ACTION TIME TICK](#)

[ACTION TIME CHANGED](#)

[ACTION TIMEZONE CHANGED](#)

[ACTION BOOT COMPLETED](#)

[ACTION PACKAGE ADDED](#)

[ACTION PACKAGE CHANGED](#)

[ACTION PACKAGE REMOVED](#)

[ACTION PACKAGE RESTARTED](#)

[ACTION PACKAGE DATA CLEARED](#)

[ACTION UID REMOVED](#)

[ACTION BATTERY CHANGED](#)

[ACTION POWER CONNECTED](#)

[ACTION POWER DISCONNECTED](#)

[ACTION SHUTDOWN](#)

Các thành phần của Intent



| Thuộc tính chính | Thuộc tính phụ |
|--|--|
| action <ul style="list-style-type: none">-tên (string) của action mà Intent sẽ yêu cầu thực hiện-có thể là action được Android định nghĩa sẵn (built-in standard action) hoặc do người lập trình tự định nghĩa | category <ul style="list-style-type: none">-thông tin về nhóm của action type <ul style="list-style-type: none">-định dạng kiểu dữ liệu (chuẩn MIME)-thường được tự động xác định |
| data <ul style="list-style-type: none">-dữ liệu mà Activity được gọi sẽ xử lý-định dạng Uri (thông qua hàm Uri.parse(data)) | component <ul style="list-style-type: none">-chỉ định cụ thể lớp sẽ thực thi Activity-khi được xác định, các thuộc tính khác trở thành không bắt buộc (optional) |
| | extras <ul style="list-style-type: none">-chứa tất cả các cặp (key,value) do ứng dụng thêm vào để truyền qua Intent (cấu trúc Bundle) |
| http://developer.android.com/reference/android/content/Intent.html | |

Các thành phần của Intent ..



- **Component name**: tên class xử lý intent (ví dụ: "com.example.project.app.MyActivity1")
- **Action**: tên các hành động mà intent yêu cầu thực hiện (ví dụ: action_view, action_call,...)
- **Data**: dữ liệu yêu cầu được xử lý, dữ liệu này thường được biểu diễn dưới dạng URI (ví dụ: "tel:216-555-1234", "https://tinhoc123.edu.vn",...)
 - Trường hợp dữ liệu phức tạp hoặc không cố định, người ta thường đẩy vào phần **extras**

Các thành phần của Intent ..



- **Type**: định dạng kiểu dữ liệu của data (dùng chuẩn MIME), thường được tự xác định bởi hệ thống
- **Category**: bổ sung thông tin cho các action của intent (ví dụ: nếu một activity có thuộc tính category là CATEGORY_LAUNCHER nghĩa là activity đó có thể khởi chạy cấp ứng dụng)
- **Extras**: dữ liệu bổ sung nếu vùng Data là chưa đủ, extras sử dụng cấu trúc **bundle** gồm các cặp (**key, value**)