LẬP TRÌNH DI ĐỘNG

Bài 5: Intent và cơ chế trao đổi dữ liệu giữa các thành phần trong android

Nhắc lại bài trước

- Các điều khiển hữu ích: ProgressBar, Progress Dialog, AutoComplete TextView, TimePicker, TimePicker Dialog, DatePicker, DatePicker Dialog, ListView, Spinner, WebView
- Quá trình xây dựng giao diện:
  1. Thiết lập giao diện trong XML
  2. Cái nào không dùng XML được thì viết trong onCreate
  3. Thiết lập dữ liệu cho điều khiển
  4. Viết các hàm xử lý sự kiện cho điều khiển
- Vài kiểu viết mã xử lý sự kiện

Nội dung

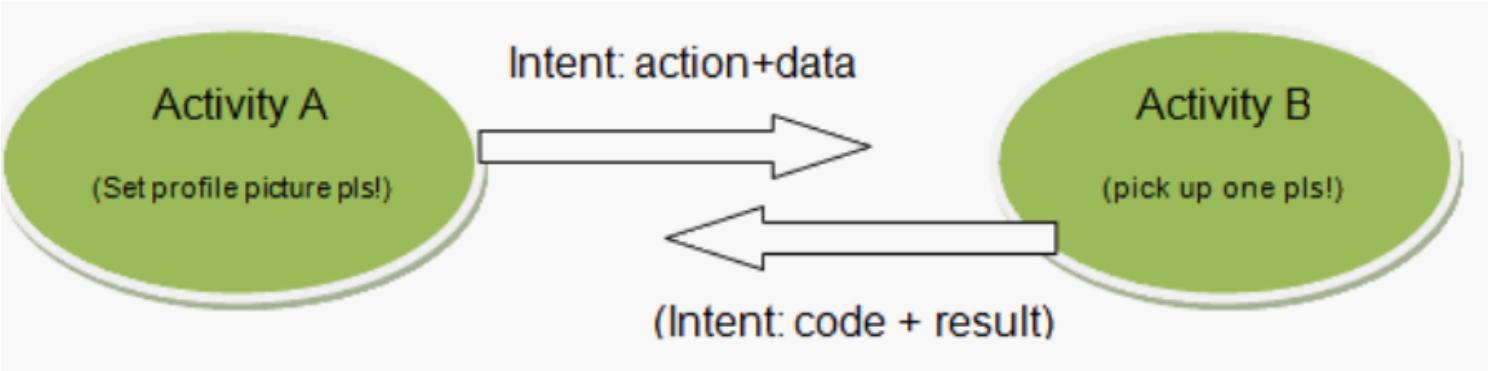
1. Giới thiệu về intent
2. Sử dụng intent để trao đổi dữ liệu
3. Intent filter
4. Intent tường minh vs ngầm định
5. Các thành phần của intent
   - Action & Data
   - Category
   - Type
   - Component
   - Extras

Phần 1

Giới thiệu về intent

Intent & Intent Service

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



Ví dụ sử dụng Intent

```
Intent x = new Intent(this, Login.class);
x.putExtra("loginname", "abcxyz");
startActivity(x);
```

mở activity Login với dữ liệu gửi kèm loginname là abcxyz

```
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(Uri.parse("http://txnam.net"));
startActivity(i);
```

mở trang web txnam.net

mở activity quay số với số 0912102165 điền sẵn

```
Intent dialIntent = new
       Intent(Intent.ACTION_DIAL, Uri.parse("tel:0912102165"));
startActivity(dialIntent);
```

Dùng Intent và nhận dữ liệu trả về

```
private static int TAKE_PICTURE = 1;
private Uri outputFileUri;
private void TakePhoto() {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        File file = new File(Environment.getExternalStorageDirectory(), "test.jpg");

        outputFileUri = Uri.fromFile(file);
        intent.putExtra(MediaStore.EXTRA_OUTPUT, outputFileUri);
        startActivityForResult(intent, TAKE_PICTURE);

}
```

bật ứng dụng camera chụp ảnh và trả về ảnh chụp

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data){

        if (requestCode == TAKE_PICTURE){
                ShowMessage(outputFileUri.toString());
        }

}
```

Phần 2

Sử dụng intent để trao đổi dữ liệu

Trao đổi intent giữa các activity

- 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

- Về phía bên 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

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

Đoạn mã minh họa việc gọi activity nhập liệu và xử lý kết quả trả về

onActivityResult: được tự động gọi khi activity nhập liệu kết thúc

```java
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 cha

Phần 3

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)

```
<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>
```

Đoạn mã XML trên 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

Phần 4

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

Hoặc

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

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 ở slide 14)
- 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 boardcast: sendBoardcast / sendOrderedBoardcast / setStickyBoardcast

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_ANSWER
ACTION_VIEW	ACTION_INSERT
ACTION_ATTACH_DATA	ACTION_DELETE
ACTION_EDIT	ACTION_RUN
ACTION_PICK	ACTION_SYNC
ACTION_CHOOSER	ACTION_PICK_ACTIVITY
ACTION_GET_CONTENT	ACTION_SEARCH
ACTION_DIAL	ACTION_WEB_SEARCH
ACTION_CALL	ACTION_FACTORY_TEST
ACTION_SEND	ACTION_SENDTO
Built-in Standard Broadcast Actions	
ACTION_TIME_TICK	ACTION_PACKAGE_RESTARTED
ACTION_TIME_CHANGED	ACTION_PACKAGE_DATA_CLEARED
ACTION_TIMEZONE_CHANGED	ACTION_UID_REMOVED
ACTION_BOOT_COMPLETED	ACTION_BATTERY_CHANGED
ACTION_PACKAGE_ADDED	ACTION_POWER_CONNECTED
ACTION_PACKAGE_CHANGED	ACTION_POWER_DISCONNECTED
ACTION_PACKAGE_REMOVED	ACTION_SHUTDOWN

Phần 5

Các thành phần của intent

Các thành phần của Intent

Thuộc tính chính	Thuộc tính phụ
action -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** -thông tin về nhóm của action
	type -định dạng kiểu dữ liệu (chuẩn MIME) -thường được tự động xác định
data -dữ liệu mà Activity được gọi sẽ xử lý -định dạng Uri (thông qua hàm Uri.parse(data))	**component** -chỉ định cụ thể lớp sẽ thực thi Activity -khi được xác định, các thuộc tính khác trở thành không bắt buộc (optional)
	extras -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 lí 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 lí, dữ liệu này thường được biểu diễn dưới dạng URI (ví dụ: "tel:216-555-1234", "http://txnam.net",…)
  - 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)