



Lập trình Android

Bài 5. *Intent*

Ngành Mạng & Thiết bị di động





Nội dung

1. Khái niệm về Intent

- Cơ chế hoạt động
- Các dạng Intent
- Xây dựng Intent
- Truy xuất Intent
- Gửi và Phản hồi Intent trong Activity

2. Intent Filter



1.1 Cơ chế hoạt động

- ❑ Intent được sử dụng để truyền tải thông điệp, yêu cầu một hành động xử lý từ thành phần được gọi.

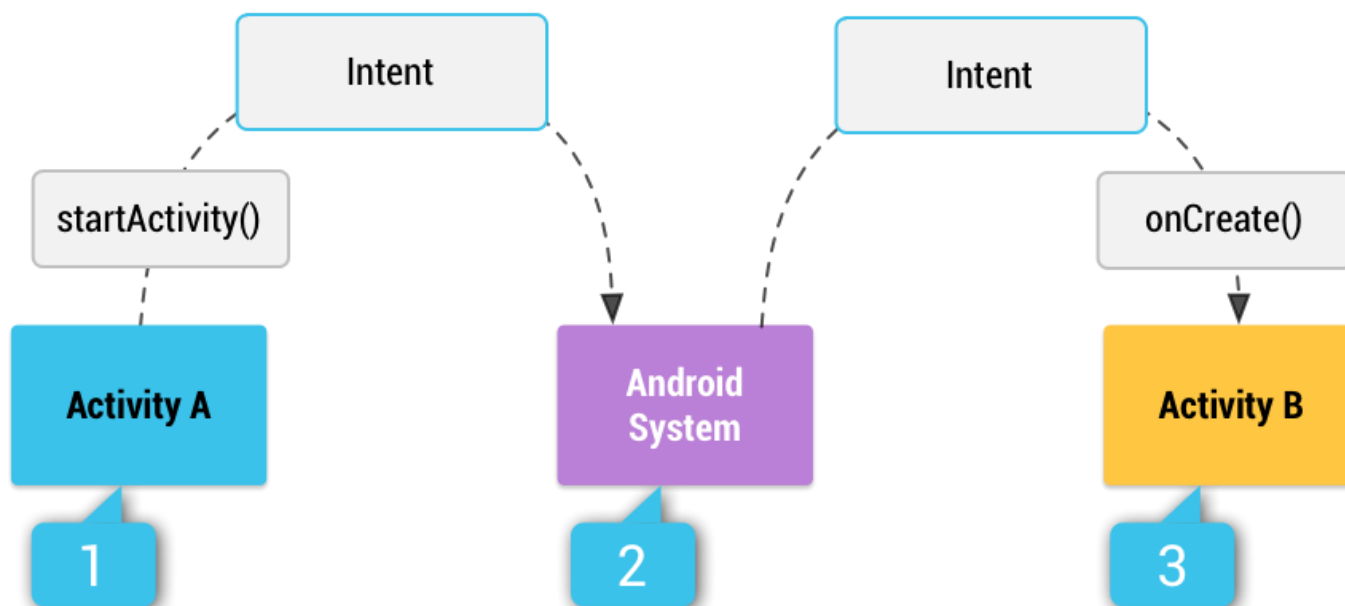
- ❑ Intent được sử dụng trong ba trường hợp chính:
 - Khởi động `Activity` thông qua phương thức `startActivity`.
 - Khởi động `Service` thông qua phương thức `startService`.
 - Chuyển thông điệp đến `BroadcastReceiver` thông qua phương thức `sendBroadcast`.



1.2 Các dạng Intent

☐ Intent được chia làm hai dạng:

- Explicit Intent: chỉ định rõ thành phần xử lý thông qua tên lớp, thường được dùng để gọi đến các thành phần trong cùng ứng dụng.
- Implicit Intent: không chỉ định rõ thành phần xử lý, thay vào đó bổ sung các thuộc tính như: mô tả hành động, dạng dữ liệu...





1.3 Xây dựng Intent

- ❑ **Đối tượng Intent khởi động các thành phần trong ứng dụng đồng thời mang các thông tin về dữ liệu được xử lý, bao gồm các thành phần sau:**
 - Component: tên thành phần nhận và xử lý Intent
 - Action: hành động yêu cầu thực thi
 - Data: dữ liệu yêu cầu nhận và xử lý
 - Category: mô tả lĩnh vực hoạt động
 - Extras: bộ key/value cho phép gửi nhận thông tin
 - Flag: biến cờ mô tả cách thức hoạt động



1.3 Xây dựng Intent

❑ Explicit Intent: chỉ cần sử dụng thuộc tính Component.

- Khai báo:

```
Intent intent = new Intent(this, <Component>);
```

- Ví dụ: khởi động Activity có tên SecondActivity từ MainActivity

```
Intent intent = new Intent(MainActivity.this, SecondActivity.class);  
startActivity(intent);
```



1.3 Xây dựng Intent

❑ **Implicit Intent: chỉ cần sử dụng thuộc tính Action.**

- Khai báo:

```
Intent intent = new Intent(<Action>);
```

- Ví dụ: khởi động Activity có thể thực hiện ACTION_VIEW.

```
Intent intent = new Intent(Intent.ACTION_VIEW);
```

```
startActivity(intent);
```



1.3 Xây dựng Intent

❑ **Action:** một số Action thường dùng trong Intent.

- ACTION_VIEW
- ACTION_DIAL
- ACTION_CALL
- ACTION_EDIT
- ACTION_DELETE
- ACTION_SEND
- ACTION_SENDTO



1.3 Xây dựng Intent

- ❑ **Data:** một dạng đường dẫn URI, cho phép trỏ đến bảng dữ liệu và truy xuất thông tin bao gồm:
 - type
 - scheme + authority + path

- ❑ **Data có thể chỉ định thông qua ba phương thức:**
 - setData
 - setType
 - setDataAndType

- ❑ **Ví dụ: thực hiện cuộc gọi thông qua dữ liệu số điện thoại**

```
Intent callPhone = new Intent(Intent.ACTION_CALL);  
callPhone.setData(Uri.parse("tel:01234-56789"));  
startActivity(callPhone);
```



1.3 Xây dựng Intent

- ❑ **Extras:** bao gồm biến Bundle chứa các giá trị bổ sung cần thiết cho thành phần nhận xử lý Intent.

- ❑ **Có hai cách gửi dữ liệu vào Intent:**
 - **Trực tiếp:**
 - Dùng phương thức `putExtra(Key, Value)` thiết lập trực tiếp vào Intent.
 - **Thông qua Bundle**
 - Tạo đối tượng Bundle, dùng phương thức `set<KDL>(Key, Value)` vào đối tượng Bundle
 - Dùng phương thức `putExtras()` gửi Bundle vào Intent.



1.3 Xây dựng Intent

❑ Extras:

- Ví dụ: gửi số nguyên x vào Intent

- Trực tiếp:

```
Intent intent = new Intent();  
intent.putExtra("SoNguyenX", x);
```

- Thông qua Bundle:

```
Intent intent = new Intent();  
Bundle bundle = new Bundle();  
bundle.putInt("SoNguyenX", x);  
intent.putExtras(bundle);
```



1.4 Truy xuất

❑ Truy xuất:

- Truy xuất dữ liệu trực tiếp Extras:
 - Dùng phương thức `get<KDL>Extra(Key, DefaultValue)` để truy xuất dữ liệu Intent.
- Thông qua Bundle
 - Dùng phương thức `getExtras()` để truy xuất đối tượng Bundle trong Intent.
 - Dùng phương thức `get<KDL>(Key, DefaultValue)` để truy xuất dữ liệu trong Bundle.



1.4 Truy xuất

❑ Truy xuất:

- Ví dụ: truy xuất số nguyên được gửi trong Intent

- Trực tiếp:

```
Intent intent = getIntent();
```

```
int soNguyenX = intent.getIntExtra("SoNguyenX", 0);
```

- Thông qua Bundle:

```
Intent intent = getIntent();
```

```
Bundle bundle = intent.getExtras();
```

```
int soNguyenX = bundle.getInt("SoNguyenX", 0);
```



1.5 Gửi và phản hồi Intent trong Activity

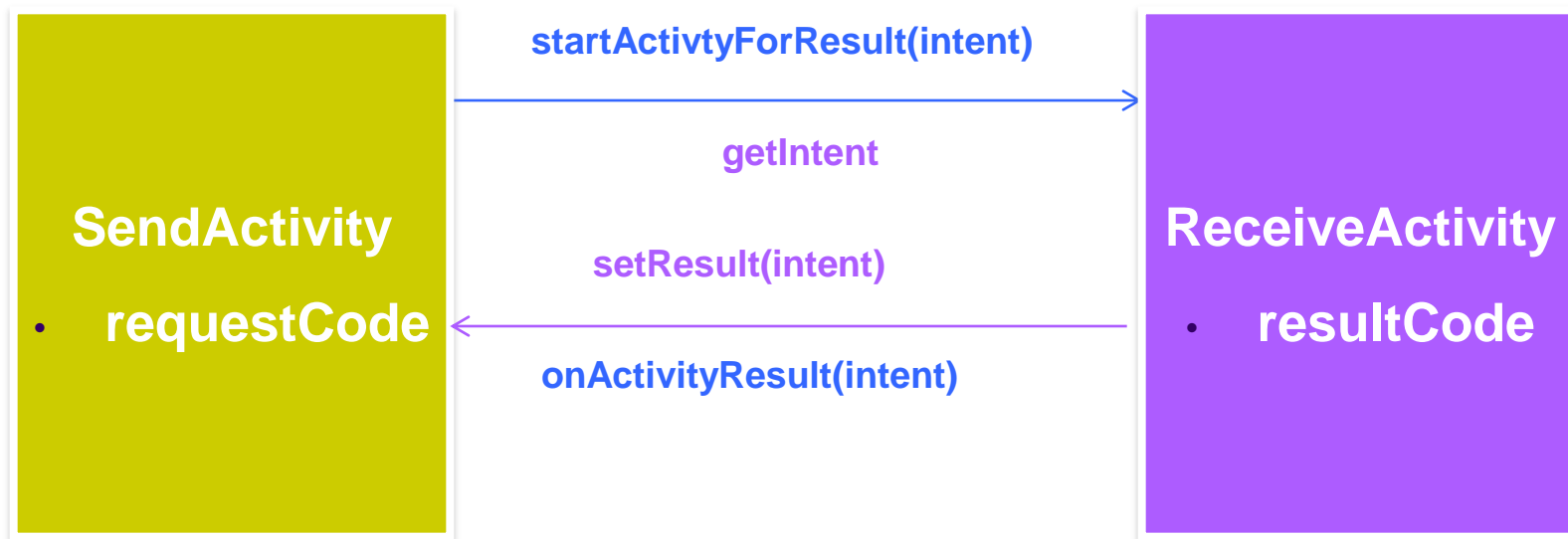
❑ Việc gửi và phản hồi Intent trong Activity được chia làm 3 bước

- Bước 1: Gửi Intent thông qua phương thức **startActivityForResult()** bao gồm 2 tham số:
 - Intent: dữ liệu cần gửi để xử lý.
 - requestCode: mã yêu cầu xử lý từ phía gửi.
- Bước 2: Nhận và xử lý Intent, sau đó xác nhận thông tin phản hồi thông qua phương thức **setResult()** trong thành phần ứng dụng phản hồi.
 - Khởi tạo đối tượng Intent, thiết lập các thuộc tính cần thiết: action, category...
 - Gửi dữ liệu phản hồi trực tiếp vào Intent hoặc thông qua biến Bundle.
 - Gọi phương thức setResult với tham số truyền vào là Intent.
- Bước 3: Gọi phương thức onActivityResult() truy xuất ba tham số:
 - requestCode: mã yêu cầu giải quyết với intent tương ứng.
 - resultCode: mã kết quả nhận về từ phía phản hồi.
 - Intent: dữ liệu nhận về từ phía phản hồi.



1.5 Gửi và phản hồi Intent trong Activity

❑ Mô hình hoạt động:





Nội dung

1. Khái niệm về Intent

2. Intent Filter

- Mô tả
- Quy tắc thiết lập
- Xây dựng IntentFilter



2.1 Mô tả

- ❑ Thực hiện mô tả cấu trúc Intent, cho phép thực hiện chỉ nhận các Intent theo đúng cấu trúc đã mô tả.

- ❑ Có thể lọc Intent theo ba thuộc tính:
 - Action
 - Data (type, scheme, authority & path)
 - Category



2.2 Qui tắc thiết lập

- ❑ **IntentFilter thực hiện lọc Intent theo thứ tự ưu tiên khi có nhiều thuộc tính được thiết lập và có những qui tắc nhất định:**
 - Nếu không thiết lập Action, chỉ nhận các Intent không có Action.
 - Nếu thiết lập thuộc tính Action và không thiết lập thuộc tính Data, chỉ cho phép lọc các Intent không có Data.
 - IntentFilter cho phép nhận các Intent có bất kỳ dữ liệu nào có liên quan đến thuộc tính Action.



2.3 Xây dựng IntentFilter

❑ Có thể khởi tạo đối tượng IntentFilter bằng 2 cách:

- Trong java Code:
 - Các hàm khởi tạo:
 - IntentFilter()
 - IntentFilter(String Action)
 - IntentFilter(String Action, URI data)
 - IntentFilter(IntentFilter o)
- Trong tập tin AndroidManifest.xml:
 - Khai báo thẻ cặp thẻ <intentfilter><intentfilter/>
 - Trong cặp thẻ có thể chứa các thẻ sau:
 - <action/>
 - <data/>
 - <category/>



2.3 Xây dựng IntentFilter

❑ Action:

- Các thuộc tính:

- `<action android:name="string" />`

- Trong đó:

- `android:name` : sử dụng các thuộc tính trong lớp `Intent.ACTION_string` hoặc tự định nghĩa chuỗi action.

- Ví dụ khai báo:

- `<action android:name="android.intent.action.MAIN" />`

- `<action android:name="com.htsi.t3h.action.ShowImage" />`



2.3 Xây dựng IntentFilter

❑ Data:

- Các thuộc tính:

- `<data android:scheme="string"`
 `android:host="string"`
 `android:port="string"`
 `android:path="string"`
 `android:pathPattern="string"`
 `android:pathPrefix="string"`
 `android:mimeType="string" />`

- Ví dụ khai báo:

```
<data android:scheme="http" android:mimeType="video/mpeg" />  
<data android:mimeType="image/*" />  
<data android:mimeType="*/*" />
```



2.3 Xây dựng IntentFilter

❑ Category:

- Các thuộc tính:

- `<category android:name="string" />`

- Trong đó:

- `android:name` : `Intent.CATEGORY_`*string*. Khai báo theo cấu trúc `android.intent.category.`*string*

- Ví dụ khai báo:

- `<category android:name="android.intent.category.DEFAULT" />`

- `<category android:name="android.intent.category.LAUNCHER" />`

