
INTENT

1. KHÁI NIỆM VỀ INTENT

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:
 - o Khởi động Activity thông qua phương thức startActivity.
 - o Khởi động Service thông qua phương thức startService.
 - o Chuyển thông điệp đến BroadcastReceiver thông qua phương thức sendBroadcast.
- Intent được chia làm hai dạng:
 - o **Explicit Intent:** (Intent tường minh) chỉ định rõ thành phần xử lý thông qua tên lớp, thường được dùng để gọi đến các thành phần trong cùng ứng dụng.
 - o **Implicit Intent:** (Intent không tường minh) không chỉ định rõ thành phần xử lý, thay vào đó bổ sung các thuộc tính như: mô tả hành động, dạng dữ liệu...

1.2. Xây dựng và truy xuất 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:
 - o **Component:** tên thành phần nhận và xử lý Intent
 - o **Action:** hành động yêu cầu thực thi
 - o **Data:** dữ liệu yêu cầu nhận và xử lý
 - o **Category:** mô tả lĩnh vực hoạt động
 - o **Extras:** bộ key/value cho phép gửi nhận thông tin
 - o **Flag:** biến cờ mô tả cách thức hoạt động
 - o **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);
```

- 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);
```

- Action: một số Action thường dùng trong Intent.
 - o ACTION_VIEW
 - o ACTION_DIAL
 - o ACTION_CALL
 - o ACTION_EDIT
 - o ACTION_DELETE
 - o ACTION_SEND
 - o ACTION_SENDTO
- Data: một dạng đường dẫn URI, cho phép trở đến bảng dữ liệu và truy xuất thông tin bao gồm:
 - o type
 - o scheme
 - o authority
 - o path
- Data có thể chỉ định thông qua ba phương thức:
 - o setData
 - o setType
 - o 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);
```

- **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:
 - o Trực tiếp: dùng phương thức putExtra(Key, Value) thiết lập trực tiếp vào Intent.
 - o Thông qua Bundle: tạo một đối tượng Bundle, dùng phương thức set<KDL>(Key, Value) vào đối tượng Bundle. Sau đó, ta dùng phương thức putExtras() gửi Bundle vào Intent.

- Ví dụ: gửi số nguyên x vào Intent
 - o Trực tiếp:

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

- o Thông qua Bundle:

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

- Truy xuất:

- o Truy xuất dữ liệu trực tiếp Extras:
 - Dùng phương thức `get<KDL>Extra(Key, DefaultValue)` để truy xuất dữ liệu Intent.
- o 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.
- o 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);
```

- Gửi và Phản hồi Intent trong Activity

- o 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.



Hình 5.1. Gửi và phản hồi Intent.

2. INTENT FILTER

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. Quy 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 như sau:
- Trong java Code:
 - Các phương thức 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/>

2.3.1. 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.nght.t3h.action.ShowImage" />
```

2.3.2. 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="*/*" />
```

2.3.3. 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" />
```