

# LẬP TRÌNH CHO CÁC THIẾT BỊ DI ĐỘNG

Bài 6: INTENT

# Intent

---

- ▶ Intents là một thành phần quan trọng trong android. Nó cho phép các thành phần ứng dụng có thể yêu cầu các hàm từ các thành phần ứng dụng android khác. Ví dụ một activity có thể chạy một activity khác ở bên ngoài để chụp ảnh.
- ▶ Intents là một objects của android.content.Intent. Intents sẽ được gửi đến hệ thống android để xác định hành động bạn muốn thực hiện, đối tượng bạn muốn xử lý.
- ▶ Intents có thể bao gồm dữ liệu thông qua Bundle. Bundle giống như một cái hộp. Bên nhận sẽ mở bundle ra nhờ key và lấy ra dữ liệu



# Intent

---

- ▶ Android intents thường được sử dụng chính:
  - ▶ Start dịch vụ
  - ▶ Gọi một activity
  - ▶ Hiển thị một trang web
  - ▶ Hiển thị danh sách liên hệ
  - ▶ Gửi một tin nhắn
  - ▶ Gọi điện thoại.



# Các loại intent

---

- ▶ Implicit Intent: là loại Intent có các Action được Android xây dựng sẵn, nó không chỉ rõ các Component xử lý (các class xử lý) mà nó sẽ cung cấp cho hệ điều hành một loạt các thông tin yêu cầu sau đó hệ điều hành sẽ đối chiếu xem trong hệ thống có bao nhiêu phần mềm khác có thể đáp ứng xử lý yêu cầu này rồi hiện ra một Dialog chứa tên của các phần mềm có thể xử lý để người dùng chọn. Nếu các phần mềm có thể xử lý thông tin chỉ có 1 thì nó sẽ được chạy luôn không qua bước hiển thị chọn phần mềm.



# Implicit Intent

---

```
Intent intent=new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse("http://www.hiepsiit.com"));  
startActivity(intent);
```



# Explicit Intent

---

- ▶ Explicit Intents là những intent đã chỉ rõ thành phần sẽ nhận và xử lý. Thông thường những intent này sẽ không cần gán bổ sung thêm các thuộc tính khác như action, data. Explicit Intent thường được sử dụng để khởi chạy các activity, hoặc service trong cùng 1 ứng dụng.

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



# Cách Truyền Dữ Liệu Qua Lại Giữa Các Activity

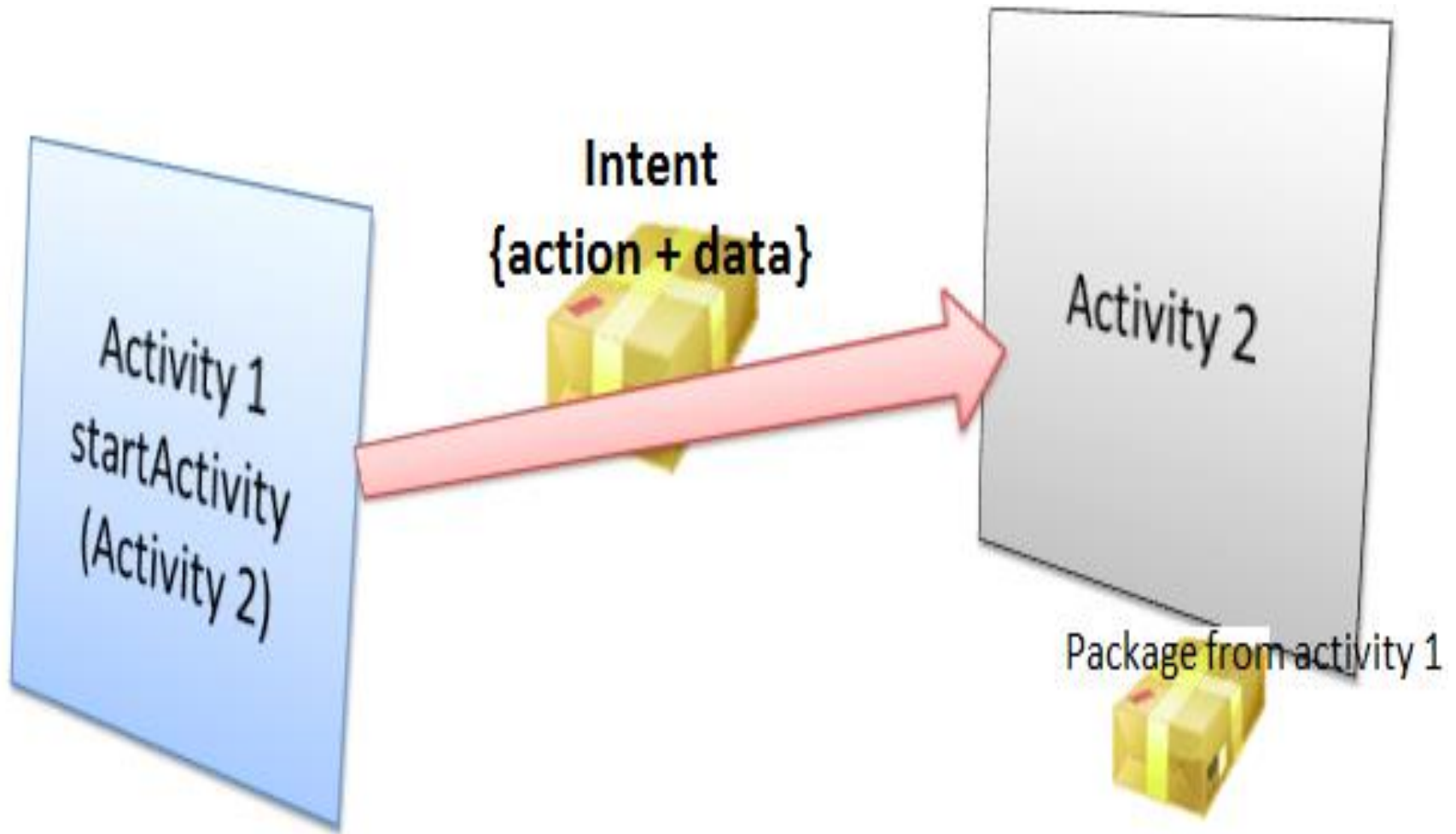
---

- ▶ Dữ liệu được “nhét” vào trong Intent và được lấy ra khỏi Intent theo các cặp dữ liệu dạng key/value. Key ở đây là một chuỗi, giúp định danh cho dữ liệu value. Nếu bạn để vào trong Intent cặp key/value nào, thì bạn phải lấy ra bởi cặp key/value đó, phải đảm bảo khai báo đúng key và lấy ra đúng kiểu dữ liệu của value khi để vào.
- ▶ Điều này tương tự như khi bạn di chuyển đi chơi xa, thì khi đóng gói hành lý của bạn, nhân viên tiếp nhận hành lý phải dán nhãn tên của bạn hay ID của bạn lên hành lý, để đảm bảo bạn lấy đúng hành lý (chính là value) khi đến nơi dựa vào nhãn tên hay ID đó (chính là key).



# Cách Truyền Dữ Liệu Qua Lại Giữa Các Activity

---





# Dùng Extra

---

- ▶ Đầu tiên, để gửi dữ liệu bằng Extra. Sau khi khai báo Intent và trước khi bạn dùng nó để kích hoạt activity nào đó, bạn có thể sử dụng các phương thức được nạp chồng của nó để gửi dữ liệu. Các phương thức đó có chung một tên là `putExtra()`.
  - ▶ Bạn nhớ là các phương thức `putExtra()` này không có s đằng sau Extra nhé. Extra có s sẽ dành cho mục Bundle dưới đây.
  - ▶ Với mỗi `putExtra()` như vậy, tham số đầu tiên chính là key đã nói ở trên. Tham số thứ hai tương tự chính là value. Phương thức này được nạp chồng làm nhiều bản để dễ dàng sử dụng từng loại value mà ta muốn.
- 



# Dùng Extra

---

```
Intent intent = new Intent(this, ContactActivity.class);  
intent.putExtra("Key_1", "Truyền một String"); //  
Truyền một String  
intent.putExtra("Key_2", 5); // Truyền một  
Int  
intent.putExtra("Key_3", true); // Truyền một  
Boolean  
startActivity(intent);
```



# Nhận Dữ Liệu

---

- ▶ Theo như ví dụ trên thì `ContactActivity` sẽ được kích hoạt với dữ liệu là ba cặp `key/value` được truyền qua. Ta có thể lấy bất cứ cặp `key/value` nào ra dùng bằng cách gọi đến `getXxxExtra()`.
- ▶ Ghi chung là `Xxx`, vì `Xxx` này sẽ được bạn thay thế bằng kiểu dữ liệu phù hợp với key bên “đóng gói”, như `getBooleanExtra()`, `getStringExtra()`, `getIntExtra()`,... Dĩ nhiên tham số `name` truyền vào phương thức này phải đúng là key bên đóng gói luôn.



# Nhận Dữ Liệu

---

```
Intent intent = getIntent();  
String value1 = intent.getStringExtra("Key_1");  
int value2 = intent.getIntExtra("Key_2", 0);  
boolean value3 = intent.getBooleanExtra("Key_3",  
false);
```



# Dùng Bundle truyền nhận dữ liệu

---

- ▶ Bundle và Extra không khác gì nhau hết. Nếu như Extra trên kia sẽ “xé lẻ” dữ liệu ra và gửi theo từng dòng. Thì Bundle sẽ giúp bạn “đóng gói” dữ liệu lại và gửi nguyên kiện.
- ▶ Tất nhiên Bundle sẽ tiện hơn trong trường hợp bạn muốn gửi cùng một bộ dữ liệu đến nhiều Activity khác nhau.



## Gửi Dữ Liệu

---

```
Intent intent = new Intent(this, ContactActivity.class);  
Bundle bundle = new Bundle();  
bundle.putString("Key_1", "Truyền một String"); //  
Truyền một String  
bundle.putInt("Key_2", 5); // Truyền một  
Int  
bundle.putBoolean("Key_3", true); // Truyền  
một Boolean  
intent.putExtras(bundle);
```



# Nhận Dữ Liệu

---

- ▶ Nếu đã gửi theo Bundle, thì bên nhận cũng sẽ nên nhận theo Bundle trước rồi mới lấy từng dữ liệu ra dùng. Để lấy Bundle ra khỏi Intent thì chúng ta có phương thức `getExtras()`
- ▶ Sau khi lấy Bundle ra khỏi Intent, việc tiếp theo sẽ gọi đến các phương thức `getXxx()` của nó. Các phương thức này của Bundle cũng giống như các phương thức `getXxxExtra()` của Extra trên kia. Chỉ khác một chỗ `getXxx()` của Bundle thường có hai phương thức nạp chồng, giúp bạn linh động hơn. Thường thì bạn nên dùng `getXxx()` với hai tham số, như vậy bạn có thể định nghĩa được giá trị mặc định cho từng phương thức khi mà nó không tìm thấy dữ liệu từ key mà bạn cung cấp, giúp tránh một số lỗi không cần thiết.



# Nhận Dữ Liệu

---

```
Intent intent = getIntent();  
Bundle bundle = intent.getExtras();  
if (bundle != null) {  
    String value1 = bundle.getString("Key_1", "");  
    int value2 = bundle.getInt("Key_2", 0);  
    boolean value3 = bundle.getBoolean("Key_3",  
false);  
}
```

