



ANDROID PROGRAMMING

LESSON 8

INTENT TRONG ANDROID

NỘI DUNG

- Android Intents
- Explicit Intents
- Implicit Intents
- Service and Broadcast Receivers

KHÁI NIỆM INTENT

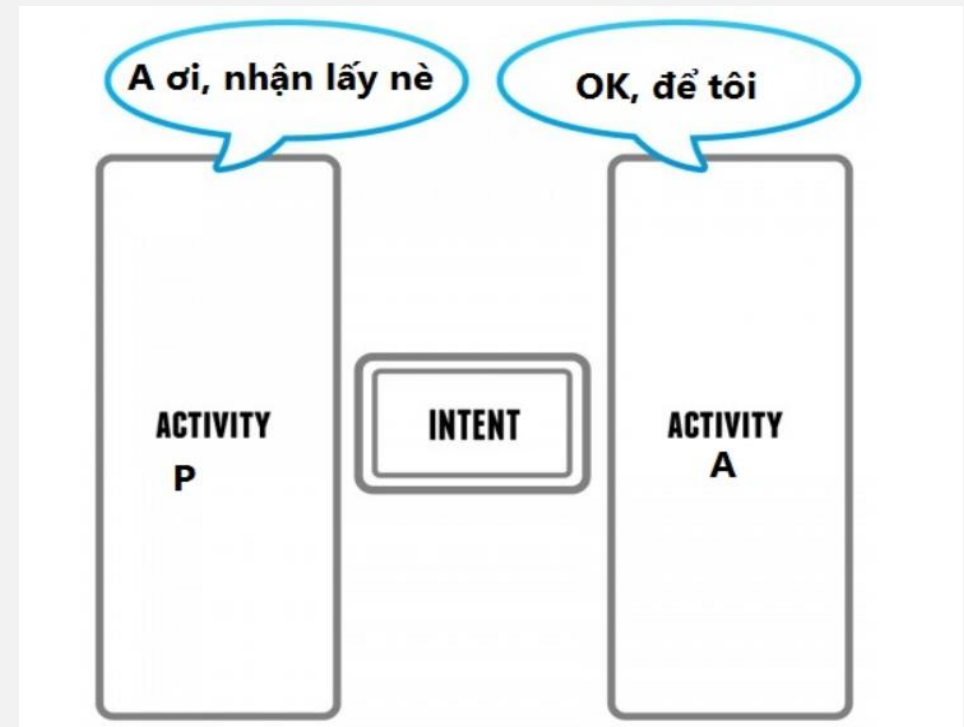
- Intent là một khái niệm trừu tượng về công việc, chức năng có thể được thực hiện bởi ứng dụng
- Intents có thể được sử dụng để:
 - Mở một Activity mới và truyền dữ liệu cho Activity đó
 - Mở một **Fragments** / Giao tiếp giữa các Fragments
 - Mở hoặc dừng một Service
 - Lắng nghe hoặc gửi thông báo tới Broadcast receiver

THÀNH PHẦN CỦA INTENT

- **Action** : thiết lập hành động của intent đó phải thực hiện
- **Data** : thiết lập kiểu dữ liệu mà intent có thể gửi đi hoặc nhận
- **Extra** : là một cặp key-value (khóa-giá trị) để truyền thông tin
- **Category** : chứa thông tin bổ sung về loại Component quản lý Intent
- **Flag** : giúp thông tin cho hệ thống Android làm thế nào để chạy một Activity và các bước tiếp theo sau khi Activity đã được khởi động.

EXPLICIT INTENT– DẠNG INTENT TƯỜNG MINH

- **Explicit Intent** : Khi tạo một đối tượng Intent, chúng ta chỉ định rõ và truyền trực tiếp tên thành phần đích vào intent



EXPLICIT INTENT

- Di chuyển giữa 2 màn hình
 - `Intent t=new Intent(MainActivity.this,SecondActivity.class);`
 - `startActivity(t);`
- Gửi dữ liệu đi kèm
 - `t.putExtra("name","To An An"); // t.putExtra(key,value);`
 - Sử dụng sau new Intent và trước khi startActivity
- Màn hình thứ 2 nhận giữ liệu
 - `Intent t=getIntent();`
 - `String name=t.getStringExtra("name"); // String name=t.getStringExtra(key);`

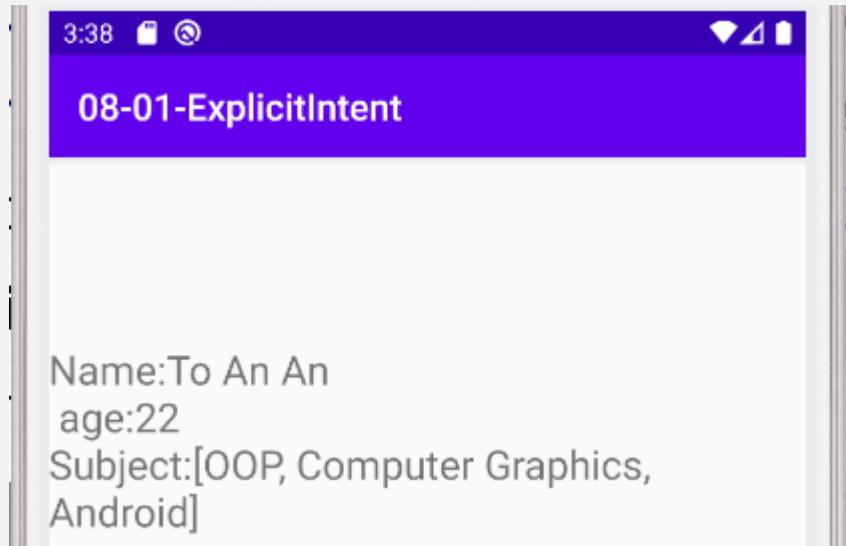
GỬI NHẬN DỮ LIỆU (01)

08-01-ExplicitIntent

Main Activity

OPEN THE SECOND ACTIVITY

```
Intent intent=new
Intent(MainActivity.this,SecondActivity.class);
String name="To An An";
intent.putExtra("name",name);
int age=22;
intent.putExtra("age",age);
String[] subject={"OOP","Computer Graphics","Android"};
intent.putExtra("subject",subject);
startActivity(intent);
```



```
tv=findViewById(R.id.result);  
Intent intent=getIntent();  
String name=intent.getStringExtra("name");  
int age=intent.getIntExtra("age",123);  
String[] subject=intent.getStringArrayExtra("subject");  
String result="Name:"+name+"\n age:"+age+"\nSubject:"+  
Arrays.toString(subject);  
tv.setText(result);
```


GỬI NHẬN OBJECT

//Gửi

```
Student s=new Student(R.drawable.chrishemsworth,"Chris  
Hemsworth",37);  
intent.putExtra("chris",s);
```

//Nhận

```
Student s=(Student) intent.getSerializableExtra("chris");  
tv.setText(result);  
img=findViewById(R.id.img);  
txtname=findViewById(R.id.tvName);  
img.setImageResource(s.getSourceImg());  
txtname.setText(s.getName());
```

MODULE LOGIN AND REGISTER (02)

08-02-LoginRegister

LOGIN FORM

Username

Password

08-02-LoginRegister

REGISTER FORM

Username

Password

08-02-LoginRegister

LOGIN FORM

Username

Password

08-02-LoginRegister

Hello

HELLO AnhTTV20 DA DANG NHAP VAO HE THONG

STARTACTIVITYFORRESULT

- sử dụng:

`startActivityForResult(Intent intent, int requestCode)`

thay vì dùng `startActivity()`, ta có thể start một Activity và sau đó nhận kết quả trả về từ Activity đó thông qua phương thức `onActivityResult()`.

STARTACTIVITYFORRESULT() :ĐƯỢC SỬ DỤNG KHI CHÚNG TA MUỐN MỞ MỘT ACTIVITY VÀ MONG MUỐN LẤY DỮ LIỆU TRẢ VỀ TỪ ACTIVITY ĐÓ.

```
public final static int REQUEST_CODE = 10000;
case R.id.btnRegister:
    Intent registerIntent = new Intent(LoginActivity.this,
        RegisterActivity.class);
    startActivityForResult(registerIntent, REQUEST_CODE);
```

- `onActivityResult()`: là phương thức xử lý kết quả trả về từ Activity mà chúng ta đã mở thông qua phương thức `startActivityForResult`.

```
protected void onActivityResult(int requestCode,
                                int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode == REQUEST_CODE) {
        if (resultCode == RESULT_OK) {
            if(data == null) {
                Toast.makeText(getApplicationContext(), "Bạn đã hủy đăng kí",
Toast.LENGTH_LONG).show();
            } else {
                u = (User) data.getSerializableExtra("Data");
                txtUsername.setText(u.getUsername());
                txtPassword.setText(u.getPassword());
            }
        }
    }
}
```

- Trong activity thứ 2 , nạp dữ liệu cho intent cần trả về activity thứ nhất và `setResult()` để android nhận diện activity nào cần nhận dữ liệu thông qua resultCode và `finish()` để hoàn thành

```
case R.id.btnRegister:  
    User user = new User(txtUsername.getText().toString(),  
                        txtPassword.getText().toString());  
    Intent intent = new Intent();  
    intent.putExtra("Data", user);  
    setResult(RESULT_OK, intent);  
    finish();  
    break;
```

MODULE LOGIN (2 BUTTONS)

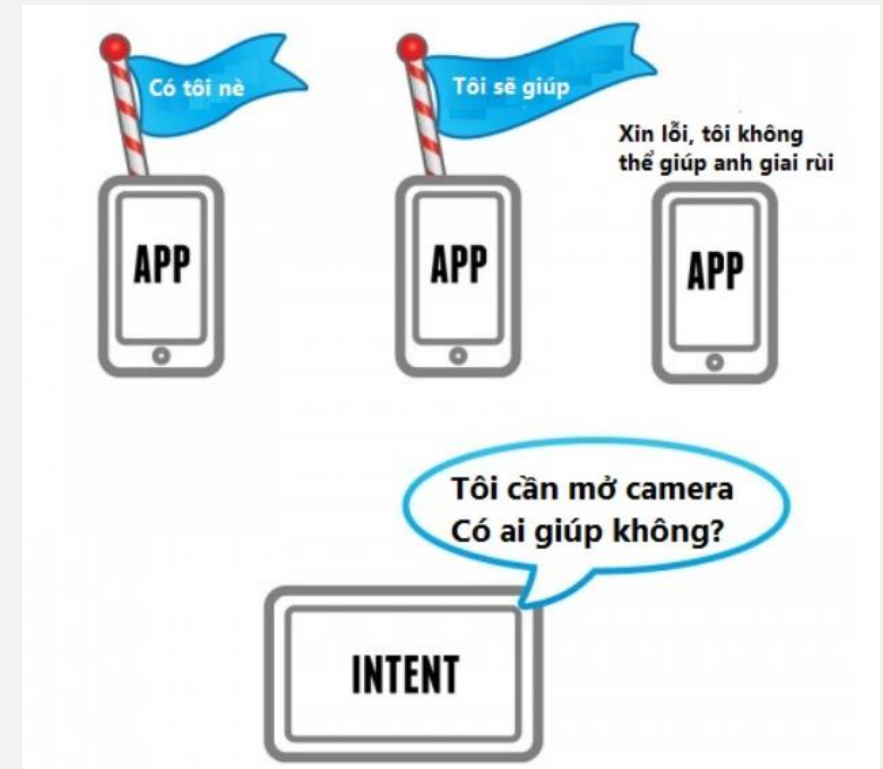
```
private final static int REQUEST_CODE = 10000;
private User u;
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnLogin:
            Intent loginIntent = new Intent(LoginActivity.this,
                MainActivity.class);
            User user = new User(txtUsername.getText().toString(),
                txtPassword.getText().toString());
            loginIntent.putExtra("account", user);
            loginIntent.putExtra("Data", u);
            startActivity(loginIntent);
            break;
        case R.id.btnRegister:
            Intent registerIntent = new Intent(LoginActivity.this,
                RegisterActivity.class);
            startActivityForResult(registerIntent, REQUEST_CODE);
    }
}
```

MODULE REGISTER (2 BUTTONS)

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.btnRegister:  
            User user = new User(txtUsername.getText().toString(),  
                                txtPassword.getText().toString());  
            Intent intent = new Intent();  
            intent.putExtra("Data", user);  
            setResult(RESULT_OK, intent);  
            finish();  
            break;  
        case R.id.btnCancel:  
            setResult(RESULT_CANCELED, null);  
            finish();  
            break;  
    }  
}  
}
```


IMPLICIT INTENT– DẠNG INTENT KHÔNG TƯỜNG MINH

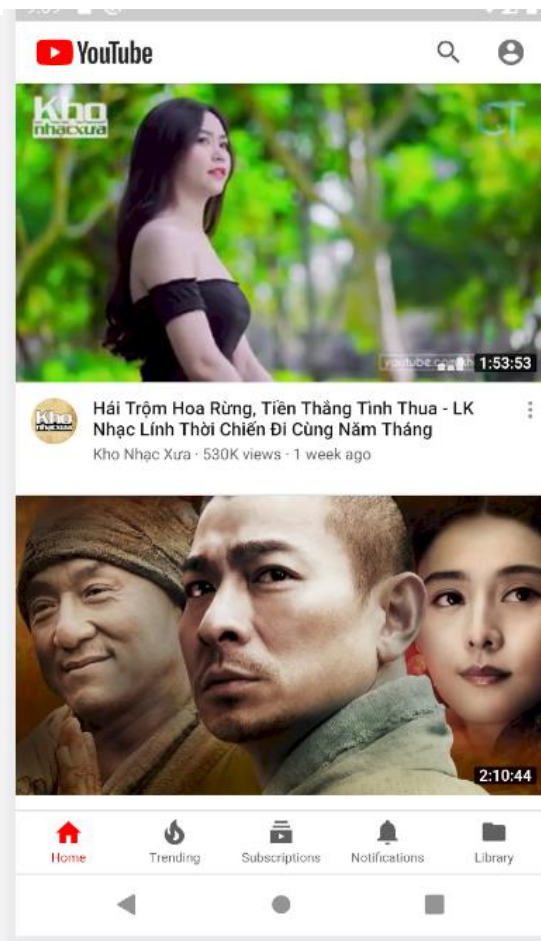
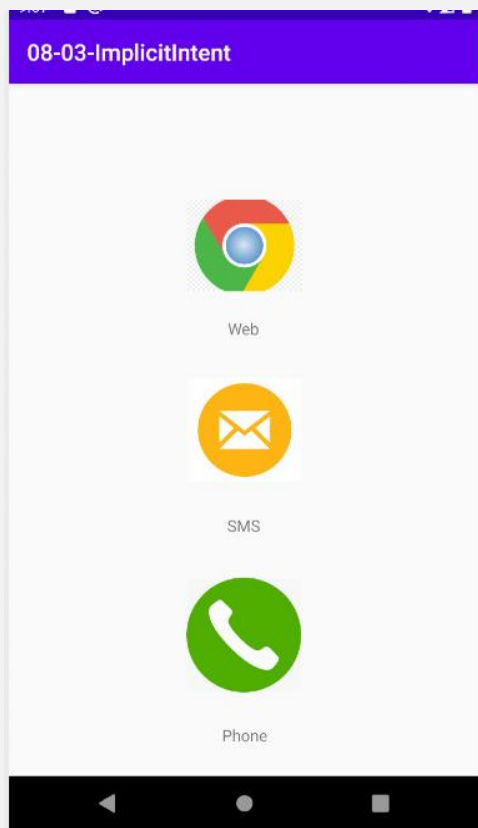
Là loại Intent có các Action được Android xây dựng sẵn, nó không chỉ rõ các Component xử lý mà cung cấp cho hệ điều hành các thông tin yêu cầu sau đó hệ điều hành sẽ đối chiếu xem trong hệ thống có bao nhiêu phần mềm khác có thể đáp ứng xử lý yêu cầu này và hiện cho người dùng lựa chọn



VÍ DỤ VỀ INTENT IMPLICIT (03)

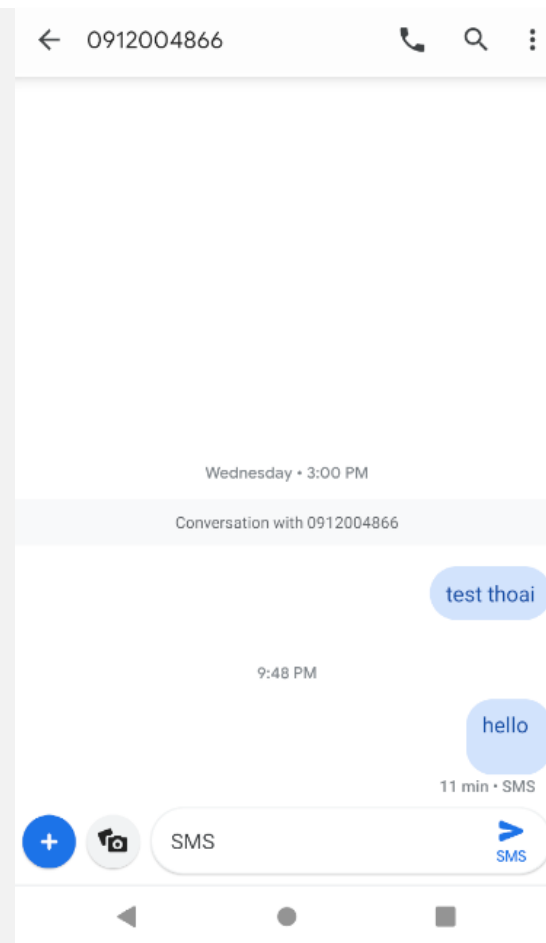
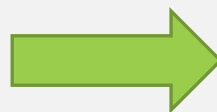
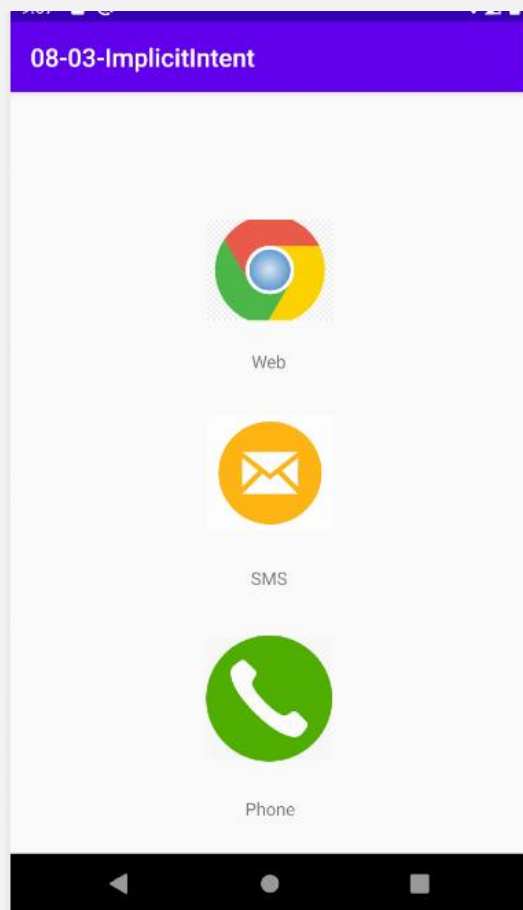
```
case R.id.imgWebview:
```

```
    Intent intentWebview = new Intent(Intent.ACTION_VIEW,  
        Uri.parse("https://www.youtube.com/"));  
    startActivity(intentWebview);  
    break;
```

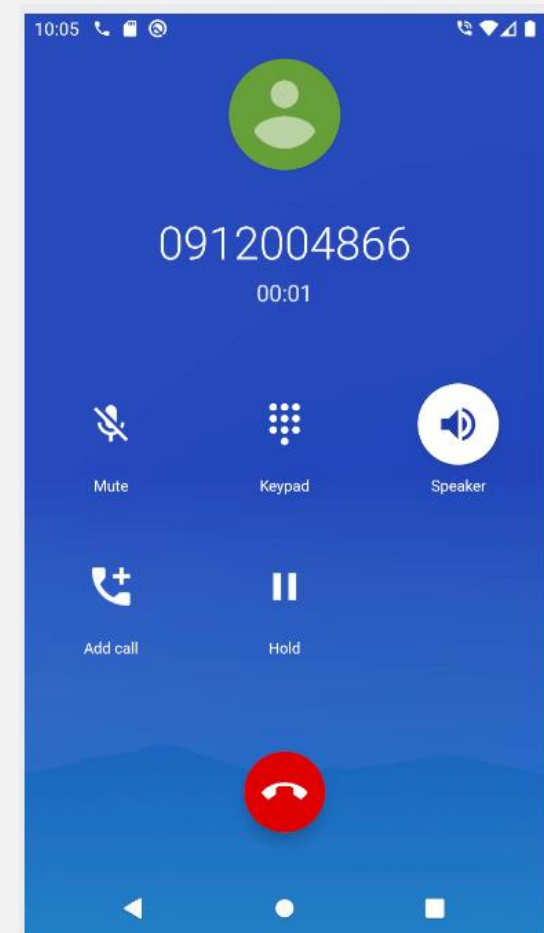
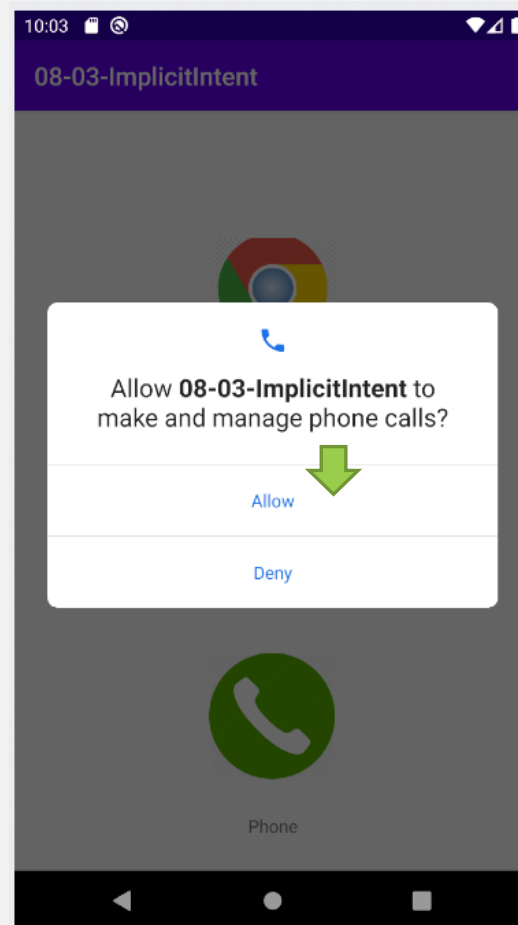
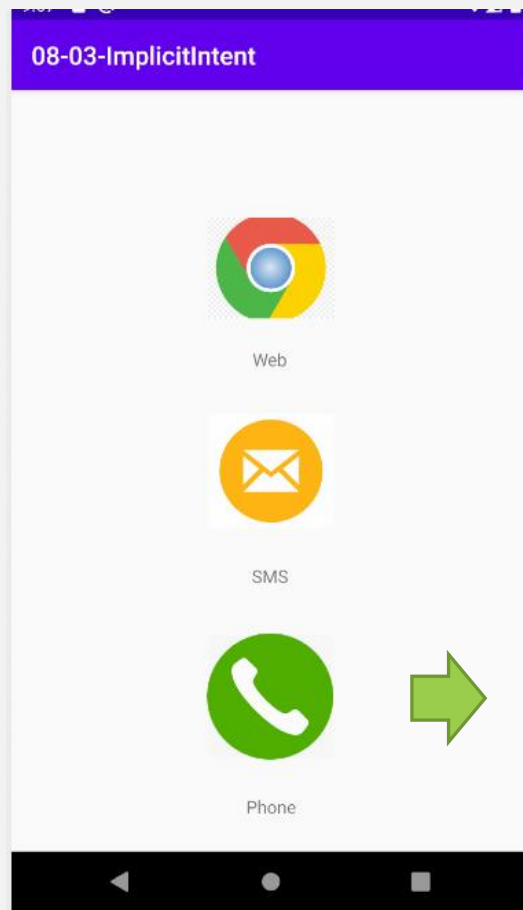


VÍ DỤ VỀ INTENT IMPLICIT

```
case R.id.imgMessage:  
    Intent intentSMS = new Intent(Intent.ACTION_VIEW);  
    intentSMS.setData(Uri.parse("sms:" + "0912004866"));  
    intentSMS.putExtra("sms_body", "SMS");  
    startActivity(intentSMS);  
    break;
```



VÍ DỤ VỀ INTENT IMPLICIT



CODE MINH HOA VỀ SỬ DỤNG INTERNET IMPLICIT ĐỂ THỰC HIỆN MỘT CUỘC GỌI TRỰC TIẾP

```
case R.id.imgPhone:
    phoneCall();
    break;
}
}
public void phoneCall(){
    if (ActivityCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.CALL_PHONE) ==
PackageManager.PERMISSION_GRANTED) {
        Intent callIntent = new Intent(Intent.ACTION_DIAL);
        callIntent.setData(Uri.parse("tel:0912004866"));
        startActivity(callIntent);
    }else{
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.CALL_PHONE}, REQUEST_CALL_PHONE);
    }
}
```

```

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions,
                                     @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if(requestCode == REQUEST_CALL_PHONE){
        if(grantResults[0] == PackageManager.PERMISSION_GRANTED){
            phoneCall();
        }
        else{
            Toast.makeText(this, "You don't assign permission.",
Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

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

```

INTENT FILTER

- Intent Filter là thành phần giúp cho hệ thống Android biết được ứng dụng có thể làm được những gì. Activity, Service và BroadCast Receiver sử dụng Intent Filter để thông báo cho hệ thống biết các dạng Implicit Intent mà nó có thể xử lý. Nói cách khác, Intent Filter là bộ lọc Intent, chỉ cho những Intent phù hợp đi qua nó.
- Trong Android, Intent Filter là một biểu thức (expression) trong file manifest của ứng dụng (ActivityManifest.xml) và nó được sử dụng để xác định loại Intent mà component muốn nhận. Trong trường hợp chúng ta tạo Intent Filter cho một activity, các ứng dụng khác sẽ có thể khởi chạy activity bằng cách gửi loại intent phù hợp, nếu không thì activity chỉ có thể được bắt đầu bằng một explicit intent.

THẺ INTENT FILTERS

- Thẻ `<intent-filter>` bắt buộc nằm trong thẻ `Activity`
- Thẻ `<intent-filter>` trong file manifest nằm trong thẻ các component tương ứng. Ta có thể xác định loại intent phù hợp thông qua ba thuộc tính sau:
- **<action>**: Nó mô tả tên của một intent action phù hợp và nó phải là chuỗi ký tự, không phải là hằng số. Có thể liệt kê nhiều hơn một action nhưng phải chứa ít nhất 1 action, nếu không nó sẽ áp dụng cho tất cả các Intent. Nếu nhiều hơn 1 Action được tìm thấy thì hệ thống Android sẽ thử chọn cái nào phù hợp trước khi khởi động Activity.

- **<category>**: mô tả tên của một loại intent được chấp nhận và nó phải là chuỗi ký tự, không phải là hằng số. Có thể liệt kê một hoặc nhiều category, hoặc có thể không có category nào. Nếu không có category nào thì hệ thống Android bỏ qua phần này. Nếu có nhiều category, từng category trong đối tượng Intent phải trùng với một trong các category của Filter.
- **<data>**: xác định loại dữ liệu sẽ phù hợp và bằng cách sử dụng một hoặc nhiều thuộc tính, chúng ta có thể chỉ định loại dữ liệu của URI (scheme, host, port, path) và loại MIME. Một đối tượng Intent chứa cả URI và kiểu dữ liệu vượt qua Filter khi và chỉ khi kiểu dữ liệu trùng với một trong các kiểu dữ liệu của Filter.

KHAI BÁO INTENT FILTER TRONG FILE MANIFEST

- file Manifest (AndroidManifest.xml)

```
<activity android:name=".MainActivity"  
    android:exported="true">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
        <data android:mimeType="text/plain"/>  
    </intent-filter>  
</activity>
```

- Thuộc tính action MAIN: chạy activity được xác định bởi action MAIN khi người dùng chạy ứng dụng bằng biểu tượng launcher.
- category LAUNCHER: biểu tượng activity này được đặt trong danh sách ứng dụng trên màn hình chính. Trong trường hợp <activity> không chỉ định biểu tượng thì hệ thống sẽ sử dụng biểu tượng từ <application>.
- Hai phần tử (MAIN, LAUNCHER) này phải được dùng cùng nhau để activity xuất hiện trong trình chạy ứng dụng.

VÍ DỤ VỀ INTENT FILTERS TRONG ANDROID

- ```
Intent email = new Intent(Intent.ACTION_SEND);
 email.setType("text/plain");
 email.putExtra(Intent.EXTRA_EMAIL, "AnhTTV20@fe.edu.vn");
 email.putExtra(Intent.EXTRA_SUBJECT, "Tham du lop hoc Andoid");
email.putExtra(Intent.EXTRA_TEXT, "Chao ban! Moi ban tham du lop hoc android");
 startActivity(Intent.createChooser(email, "Chọn một email từ..."));
```
- ```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <action android:name="android.intent.action.SEND"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
</intent-filter>
```

OPEN WEB

- `intent.setAction(Intent.ACTION_VIEW);`
`intent.setData(Uri.parse("https://www.youtube.com/channel/UCj5v57mkm-oHHVFNQZ8-DNg/videos"));`
`startActivity(intent);`
- Viết vào file `:AndroidManifest.xml`
- `<intent-filter tools:ignore="AppLinkUrlError">`
 `<action android:name="android.intent.action.VIEW"/>`
 `<category android:name="android.intent.category.DEFAULT"/>`
 `<data android:scheme="https"/>`
`</intent-filter>`

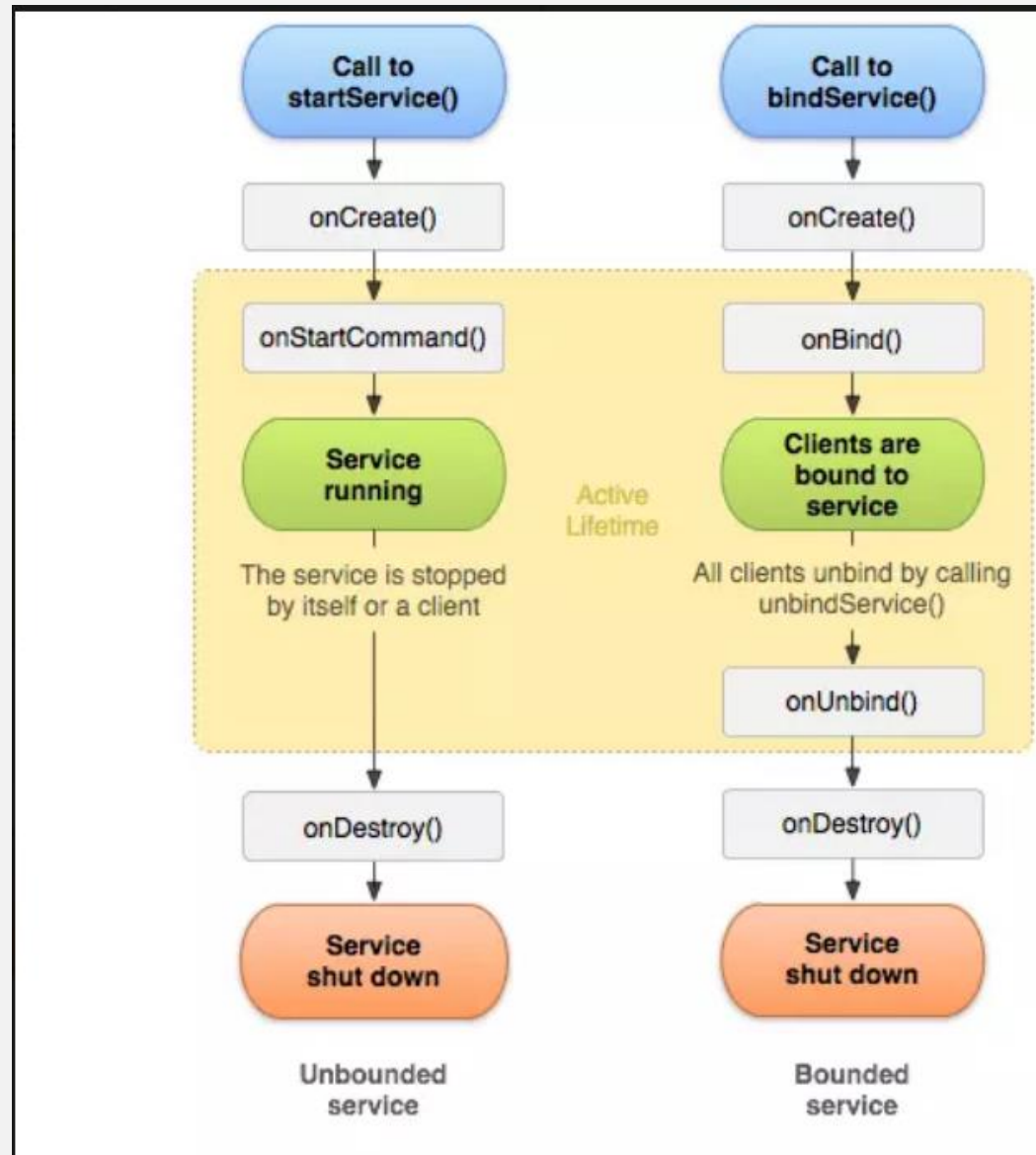
SEND MESSAGE

- `intent.setAction(Intent.ACTION_VIEW);`
`intent.setData(Uri.parse("sms:0912004866"));`
`intent.putExtra("sms_body","ban co tham gia khong");`
`startActivity(intent);`
- Viết vào file :AndroidManifest.xml
- `<intent-filter>`
 - `<action android:name="android.intent.action.VIEW"/>`
 - `<action android:name="android.intent.action.SENDTO"/>`
 - `<category android:name="android.intent.category.DEFAULT"/>`
 - `<data android:scheme="sms"/>``</intent-filter>`

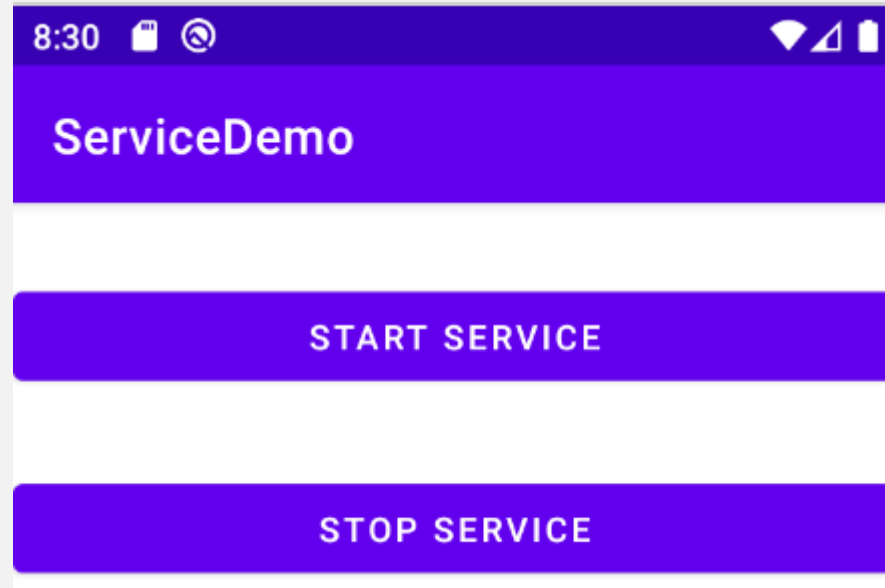
SERVICE

- Tiến trình ngầm (Service) hay các Broadcast Receiver.
- Service là thành phần ứng dụng có thể thực hiện các thao tác đòi hỏi tốn nhiều thời gian và tài nguyên
- Service không cung cấp giao diện cho người dùng
- Service chạy ngầm để thực hiện các công việc như nghe nhạc, thực hiện ghi và đọc file, hoặc tương tác với Content Provider
- Có 2 loại: Foreground Service (music player sẽ sử dụng service để chơi nhạc) và Background Service (tạo báo thức, thông báo báo thức cho người dùng bằng âm thanh và notification)

VÒNG ĐỜI CỦA SERVICE



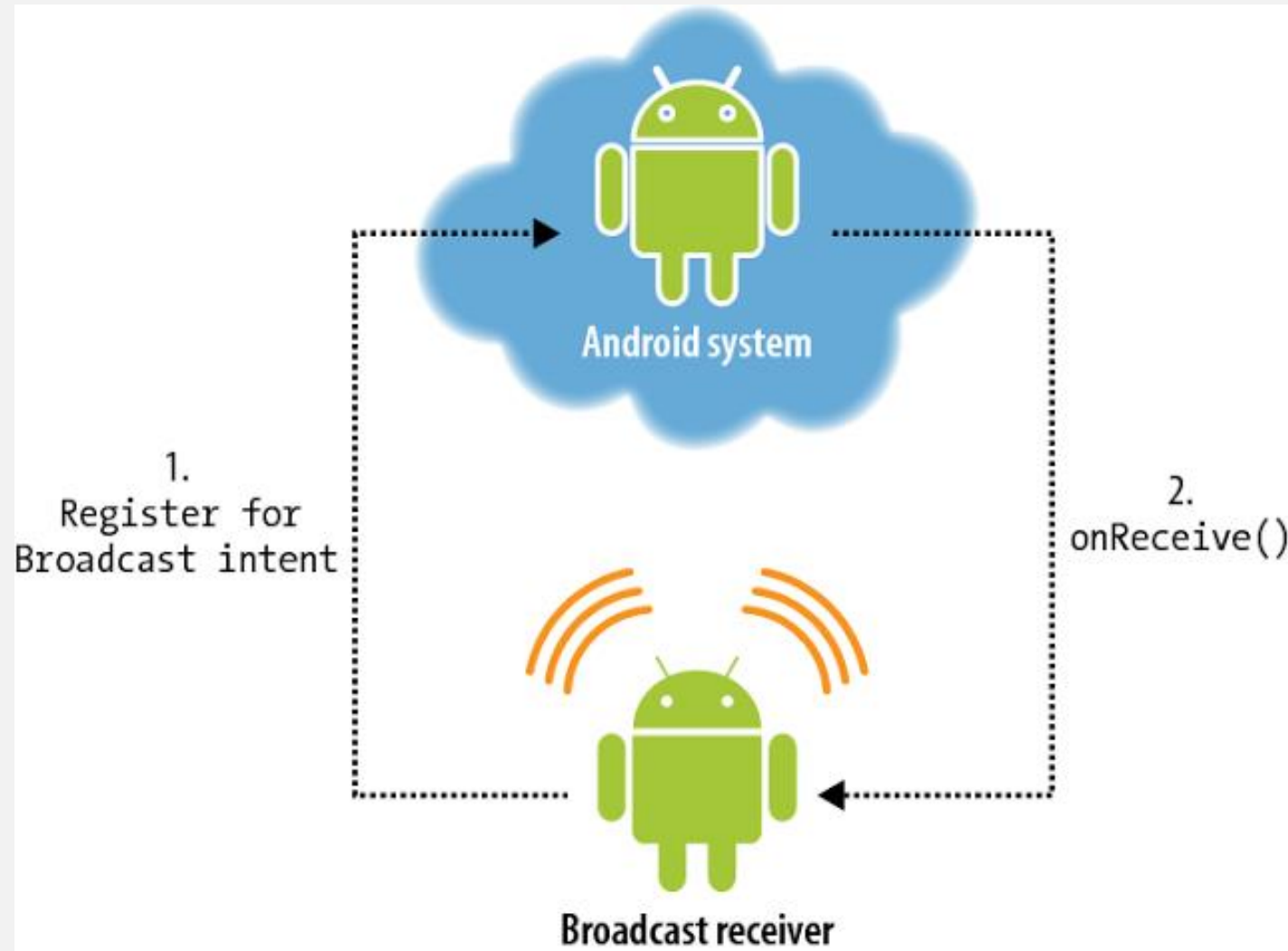
VÍ DỤ



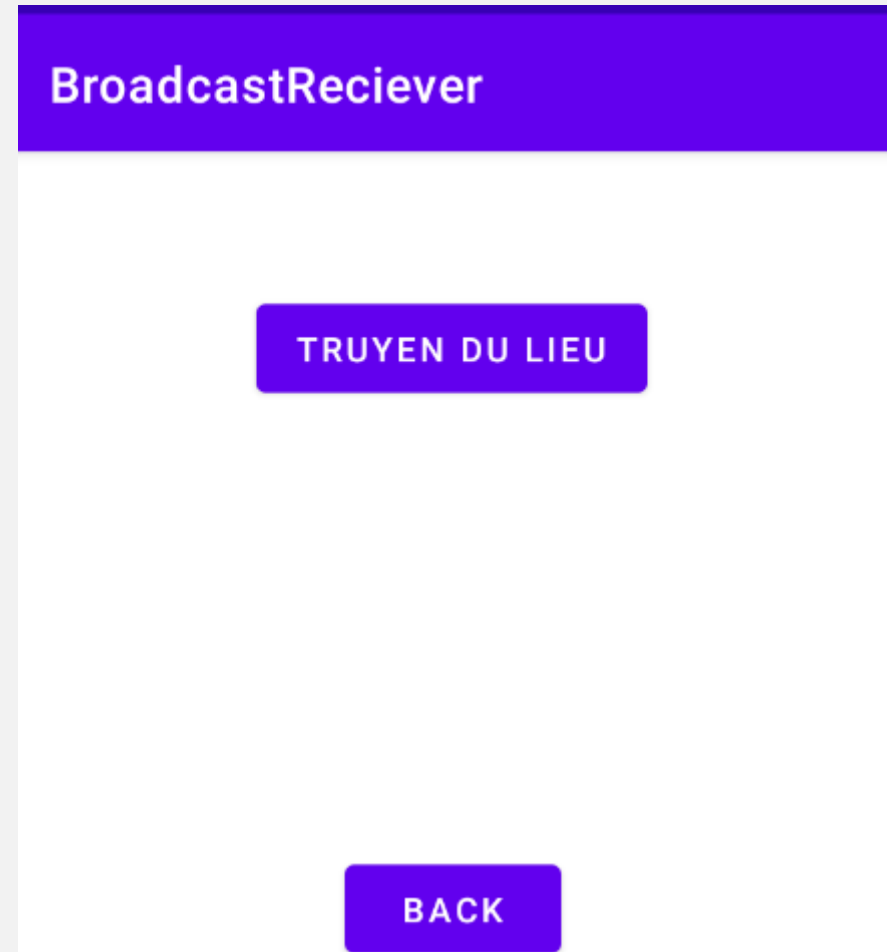
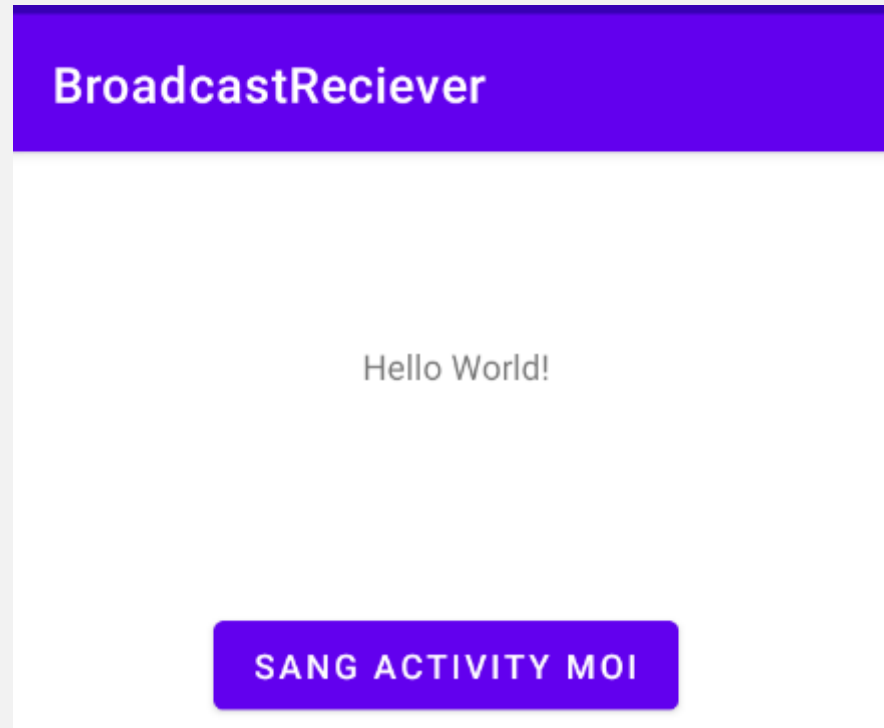
BROADCAST RECEIVER

- Chức năng dùng để nhận các sự kiện mà các ứng dụng hoặc hệ thống phát đi
- Có 2 cách phát-nhận:
 - Không có thứ tự: receiver nào đủ điều kiện thì nhận, không phân biệt và tách rời nhau
 - Có thứ tự: receiver nào đăng ký ưu tiên hơn thì nhận trước, và có thể truyền thêm thông tin xử lý cho các receiver sau
- Vòng đời của Broadcast Receiver gồm duy nhất một phương thức `onReceive()`
 - Khi có sự kiện mà Broadcast Receiver đã đăng ký nhận được phát đi, thì phương thức `onReceive()` của `BroadcastReceiver` đó sẽ được gọi
 - Sau khi thực hiện xong phương thức này, vòng đời của Broadcast Receiver kết thúc

VÒNG ĐỜI CỦA BROADCASTRECEIVER



VÍ DỤ



THE END !

Thank you!