



Trang chủ

- [PHP](#)
- [Liên hệ](#)



- [HTML](#)
- [CSS](#)
- [JavaScript](#)
- [PHP](#)
- [Zend Framework](#)
- [Server](#)
- [Java](#)
- [Android](#)
- [Liên hệ](#)

Lập trình Android - Java §1) [Android Studio - Ứng dụng đầu tiên](#) §2) [Thành phần ứng dụng](#) §3) [Vòng đời Activity](#) §4) [Activity cơ bản](#) §5) [Lưu và phục hồi trạng thái](#) §6) [View, ViewGroup, Layout](#) §7) [TextView](#) §8) [Button](#) §9) [CheckBox](#) §10) [Switch / SwitchCompat](#) §11) [ToggleButton](#) §12) [RadioButton](#) §13) [TextClock](#) §14) [EditText](#) §15) [ImageView / ImageButton](#) §16) [FrameLayout](#) §17) [LinearLayout](#) §18) [ListView](#) §19) [ConstraintLayout](#) §20) [TableLayout](#) §21) [RelativeLayout](#) §22) [Drawable - shape](#) §23) [BitmapDrawable](#) §24) [VectorDrawable](#) §25) [NinePatchDrawable](#) §26) [LayerDrawable](#) §27) [StateListDrawable](#) §28) [ActionBar - Toolbar](#) §29) [Snackbar](#) §30) [FAB - FloatingActionButton](#) §31) [RecyclerView](#) §32) [CoordinatorLayout 1](#) §33) [CoordinatorLayout 2](#) §34) [Broadcast Intent](#) §35) [Theme Android \(1\)](#) §36) [Theme Android \(2\)](#) §37) [SQLite \(1\)](#) §38) [SQLite \(2\)](#) §39) [Màn hình Ngang - Đọc](#)

- [Lập trình PHP](#)
 - [PHP Cơ bản](#)
 - [SPL](#)
 - [PSR](#)
- [Lập trình ứng dụng iOS - Swift](#)
- [Lập trình Dart - Flutter](#)
- [Lập trình C# \(C Sharp\)](#)
- [Xenforo](#)
- [Zend Framework](#)
 - [Expressive](#)
- [Server](#)
 - [MySql Server](#)
 - [Apache](#)
 - [PHP](#)
- [HTML](#)
- [Javascript](#)
 - [JQuery](#)
 - [TypeScript - Angular](#)
- [CSS](#)
 - [Sử dụng SASS / SCSS](#)
 - [Bootstrap - CSS Framework](#)
- [SQL](#)
 - [SQL Server \(.NET Framework - C#\)](#)
- [Java](#)
 - [Android Java](#)
- [Thuật ngữ - Các vấn đề cơ bản](#)
- [Tools](#)
 - [Git và GitHub](#)
 - [Grunt](#)
 - [Docker](#)
 - [macOS](#)
 - [English Study](#)
- [Tin tức công nghệ](#)
- [Java](#)
- [Android Java](#)

Cơ bản về cách tạo và sử dụng Activity trong Android

Tổng quát về Activity, AppCompatActivity cách chèn giao diện layout, khai báo manifest, các gọi một Activity, truyền và nhận dữ liệu giữa các Activity bằng đối tượng Intent

Cơ bản về Activity

Một **Activity** bạn tạo ra bằng cách kế thừa (extends) lớp **android.app.Activity** (hoặc lớp *AppCompatActivity*, ta sẽ dùng cách này, xem chú thích phần dưới) mà khi kích hoạt nó sẽ tạo ra một cửa sổ nơi chúng ta sẽ chèn giao diện (UI) vào bằng phương thức **setContentView(View)**

Như trình bày trong [vòng đời Activity Android](#) có một số giai đoạn. Để **Activity** hoạt động hoạt động tối thiểu bạn cần nạp chồng (overridden) phương thức **onCreate(Bundle)**, nó là phương thức được gọi ở giai đoạn đầu tiên của chu kỳ vòng đời Activity, chính vì thế đây là chỗ mà chúng ta sẽ nạp giao diện cho nó bằng phương thức **setContentView(int)** hoặc **setContentView(View)**

Bạn cũng có thể cần nạp chồng phương thức **onPause()**, phương thức này gọi khi Activity chuyển sang trạng thái tạm dừng tức là lúc giao diện của nó không còn xuất hiện nữa (ví dụ khi kích hoạt một Activity khác, Activity khác nó hoạt động và che hết màn hình), code trong phương thức này thường để lưu lại các thông tin (vì Activity có thể không được kích hoạt lại)

Ví dụ tạo một Activity tên là: Lesson2Activity

```
public class Lesson2Activity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.lesson2_activity);

        //Các khởi tạo khác nếu cần thiết
    }

    @Override
    protected void onPause() {
        super.onPause();

        //Code to save data
    }
}
```

AJ-03: Tổng quan chạy, tra...



Trong đoạn code trên có **setContentView(R.layout.lesson2_activity);** thì đã yêu cầu nạp UI có tên là **lesson2_activity**, vậy ta cần tạo UI này, nó tương ứng với file `res/layout/lesson2_activity.xml`, ví dụ tạo ra mẫu đơn giản gồm có một **TextView** hiển thị một đoạn văn, và một nút bấm **Button**:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <TextView
        android:id="@+id/noidung"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="8dp"
        android:background="#2aab1194"
        android:padding="5dp"
        android:text="Đây là một đoạn văn trong View của layout:
                    lesson2_activiy. Lập trình Android bài 2."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/editbutton"
        android:text="Edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent" />

</android.support.constraint.ConstraintLayout>
```

Trong phần Layout này, ngoài phần tử gốc là `ConstraintLayout`, có sử dụng phần tử là **TextView** và gán một chỉ số ID cho nó tên là **noidung** với cú pháp thuộc tính `android:id="@+id/noidung"` và phần tử **Button** có gán ID là **editbutton**

Để gán **id** cho các View ở file xml, dùng cú pháp: **android:id="@+id/tên_id"**. Giá trị của chỉ số đó tự động phát sinh bởi SDK, muốn lấy giá trị đó dùng cú pháp **R.id.tên_id**

Khi mà Activity `Lesson2Activity` nạp xong layout bởi hàm **setContentView(int)** thì nó đã tạo ra đối tượng lớp `TextView`, `Button` thông số mô tả trong layout. Để lấy được đối tượng này ở phần code Java dùng lệnh **findViewById(int)**, bạn có thể cập nhật lại `onCreate` ở trên như sau để lấy được hai đối tượng View trên.

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.lesson2_activity);

    TextView textView = findViewById(R.id.noidung);
    Button editbutton = findViewById(R.id.editbutton);
}
```

Để tìm ra View theo id, dùng phương thức **findViewById(id_của_view)**, nó sẽ tìm trên toàn bộ giao diện nạp vào cho Active, nếu thấy View nó sẽ trả về đối tượng View, nếu không trả về **null**

Bắt sự kiện khi bấm vào View

Tất cả các View đều có khả năng lắng nghe các sự kiện chạm / vuốt trên màn hình, nếu muốn View nào đó bắt sự kiện click (bấm + nhả nhanh) thì sử dụng phương thức **setOnClickListener(listener)** của View, trong đó **listener** là đối tượng lớp **View.OnClickListener** (nạp chồng phương thức `onClick(View view)` là phương thức gọi khi có sự kiện xảy ra)

Ví dụ sửa lại `onCreate` để bắt sự kiện khi bấm vào nút `editbutton`, khi bấm vào thì hiện dòng chữ thông báo **Vừa bấm vào Edit**

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.lesson2_activity);

    TextView textView = findViewById(R.id.noidung);
    Button editbutton = findViewById(R.id.editbutton);

    //Tạo listener cho sự kiện Click
    View.OnClickListener listenerEditbutton = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
```

```
        Toast.makeText(view.getContext(), "Vừa bấm vào Edit",
            Toast.LENGTH_SHORT).show();
    }
};
//Gán Listener cho editbutton
editbutton.setOnClickListener(listenerEditbutton);
}
```

Để popup một box nhỏ hiển thị thông báo thì dùng **Toast.makeText(context,thongbao,Toast.LENGTH_SHORT).show();** trong đó context là đối tượng Context, Active là một Context, nếu là View thì lấy Context của nó bằng View.getContext(), thongbao là dòng chữ String hoặc CharSequence

Cấu hình Activity trong AndroidManifest.xml

Để một Activity có thể mở ra được (chạy được) bằng phương thức **Context.startActivity();** hoặc khởi chạy khi mở ứng dụng thì phải khai báo nó trong AndroidManifest.xml.

Mỗi Activity khai báo trong manifest đều nằm trong phần tử **application** với cú pháp

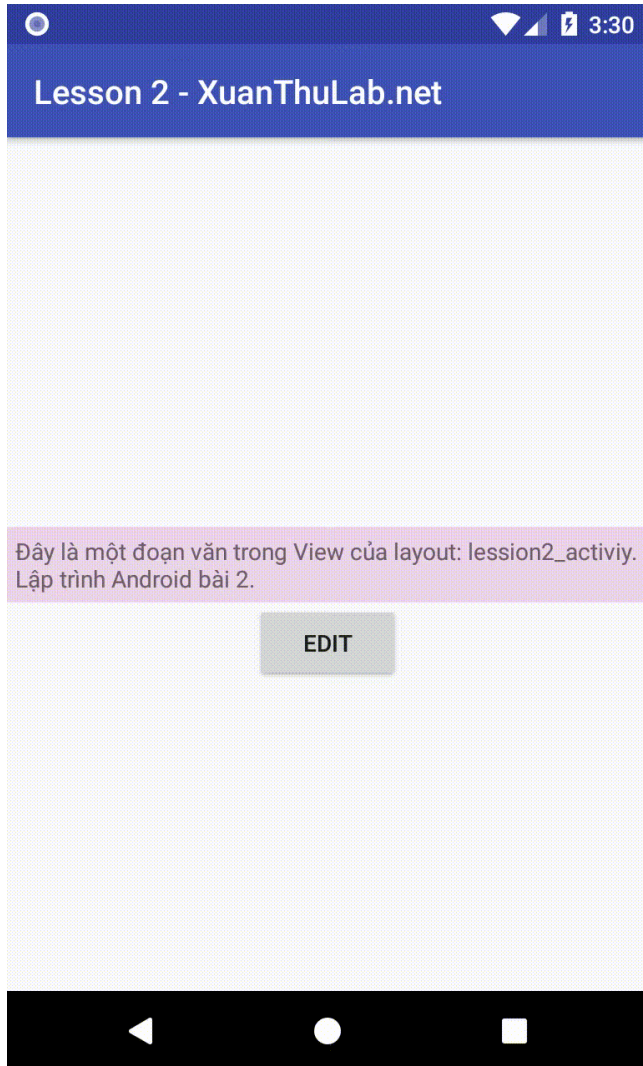
```
<!-- .... --!>
<activity android:name="Lớp_Activity">
    <!-- các tham số con --!>
</activity>
<!-- .... --!>
```

Ngoài thuộc tính **android:name** như trên, tùy thuộc mục đích còn có nhiều thuộc tính khác kế thừa từ Theme, hoặc được thiết lập trực tiếp như: android:screenOrientation, android:label, android:icon, android:permission ... (xem thêm [Activity Manifest](#))

Để thiết lập Activity là Activity mặc định (chạy đầu tiên khi mở ứng dụng thì thêm intent-filter như sau vào trong phần tử activity đó, ví dụ:

```
<!-- .... --!>
<activity android:name="Lớp_Activity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<!-- .... --!>
```

Chạy thử:



Một số điều cơ bản cho phần này liên quan đến Activity

- Để tạo ra một Activity thì cần tạo lớp kế thừa từ Activity (android.app.Activity), AppCompatActivity (android.support.v7.app.AppCompatActivity) nhưng nên triển khai từ AppCompatActivity vì đây cũng chính là một lớp kế thừa từ lớp Activity nhưng nó định nghĩa trong [Support Library](#), thư viện này hỗ trợ nhiều tính năng hiện đại chạy được trên Android Version cũ (tương thích ngược)
- Activity phải nạp chồng phương thức onCreate(), và để chèn giao diện vào Activity thì trong **onCreate** gọi phương thức `setContentView(R.layout.tên_layout);`
- Để tìm một View trong Activity dùng phương thức `findViewById(id_của_view);`
- Trong quá trình lập trình cho Android, để lấy được các thành phần tài nguyên từ ID, layout ... phải dùng đến đối tượng Context, các Activity chính là một Context vì nó thực ra mở rộng từ Context, trong các View lấy Context bằng phương thức `View.getContext()`

Sử dụng Intent để chạy Activity

Chúng ta đều biết hệ điều hành Android được thiết kế cho các thiết bị cầm tay như điện thoại thông minh, máy tính bảng, đồng hồ thông minh ... đặc điểm chung của các thiết bị này so với Desktop, Laptop ... đó là tài nguyên hệ thống thấp hơn nhiều như bộ nhớ ít hơn, tốc độ CPU chậm hơn ... nên Android có cơ chế để đảm bảo các tài nguyên đó được quản lý và thu hồi hợp lý nhất.

Một trong các cơ chế đó là việc quản lý các Activity ứng dụng, khi giao diện của Activity không còn xuất hiện trên màn hình nó chuyển sang chế độ Pause, và Android có thể hủy Activity đó bất kể lúc nào nếu cần để thu hồi tài nguyên, thứ hai là giữa các Activity không bao giờ truy cập trực tiếp với nhau (Activity này không thể gọi phương thức, hay truy cập dữ liệu của Activity khác).

Do giữa các Activity không được liên lạc trực tiếp với nhau, nên để từ Activity này chạy một Activity khác, trao đổi dữ liệu giữa các Activity phải dùng tới một đối tượng có tên là **Intent**

Intent có nhiều mục đích, ở đây chỉ thảo luận việc dùng nó để từ một Activity này mở ra một Activity khác, dùng nó để gửi dữ liệu trả về. Giả sử có một Activity tên lớp là **UpdateActivity**, để từ một Activity nào đó mở được UpdateActivity thì tạo ra một đối tượng Intent thiết lập tham số phù hợp rồi gọi **startActivity(intent)** để gửi yêu cầu và dữ liệu mô tả trong Intent cho hệ thống Android, và hệ thống kích hoạt Activity phù hợp

Code khởi chạy Activity thông thường

```
Intent intent = new Intent();

//Gán context và tên Lớp Activity cần chạy
intent.setClass(view.getContext(), UpdateActivity.class);

//Nếu có truyền tham số cho Activity
intent.putExtra("dulieu1", 345);
intent.putExtra("dulieu2", "Một chuỗi ...");

//Gửi Intent cho hệ thống Android để kích hoạt Activity
startActivity(intent);
//Muốn Activity thứ nhất kết thúc thì thêm finish();
```

Khi hệ thống Android tạo ra Activity thứ 2 (UpdateActivity) thì nó cũng bắt đầu bằng hàm onCreate(); Trong hàm này có thể lấy tham số Activity thứ nhất truyền qua bằng cách truy cập vào đối tượng Intent của Activity thứ 2, lấy Intent chứa dữ liệu bằng phương thức **getIntent()**; Có Intent rồi thì chỉ việc đọc dữ liệu bằng các phương thức tương ứng với kiểu dữ liệu theo tên như: **intent.getStringExtra("tendulieu")**; **intent.getIntExtra("tendulieu", 0)**; ...

Ví dụ đọc dữ liệu trong onCreate

```
Intent intent = getIntent();
int dulieu1 = intent.getIntExtra("dulieu1", 0);
String dulieu2 = intent.getStringExtra("dulieu2");
```


Thực hành thiết kế UpdateActivity

UpdateActivity mục đích có chức năng để cập nhật nội dung văn bản, nó mở ra khi bấm vào nút Edit trên Lesson2Activity

res\layout\update_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UpdateActivity"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <EditText
        android:id="@+id/noidung_edit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="8dp"
        android:lines="10"
        android:padding="5dp"
        android:gravity="top"
        android:background="#e8e5ea"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/ok_button"
        android:text="Save"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/noidung_edit"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        />

</android.support.constraint.ConstraintLayout>
```

UpdateActivity.java

```
package net.xuanthulab.firstapp;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
```

```

import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class UpdateActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.update_activity);

        EditText noidung_edit = findViewById(R.id.noidung_edit);

        Intent intent = getIntent();
        String noidung = intent.getStringExtra("noidung");
        noidung_edit.setText(noidung);

        Button ok_button = findViewById(R.id.ok_button);
        //Bắt sự kiện khi bấm nút Save
        ok_button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                finish(); //Kết thúc Activity (sẽ quay trở về Activity trước nó)
                Toast.makeText(view.getContext(), "Save", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

Thêm vào manifest bên trong phần tử application mô tả về UpdateActivity

```
<activity android:name=".UpdateActivity" android:label="Edit content"/>
```

Trong onCreate của Lesson2Activity thêm vào đoạn mã để khi bấm vào nút **Edit** sẽ mở UpdateActivity

```

Button edibutton = findViewById(R.id.editbutton);
//Tạo listener cho sự kiện Click
View.OnClickListener listenerEdibutton = new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent intent = new Intent();

        //Gán context và tên Lớp Activity cần chạy
        intent.setClass(view.getContext(), UpdateActivity.class);
    }
}

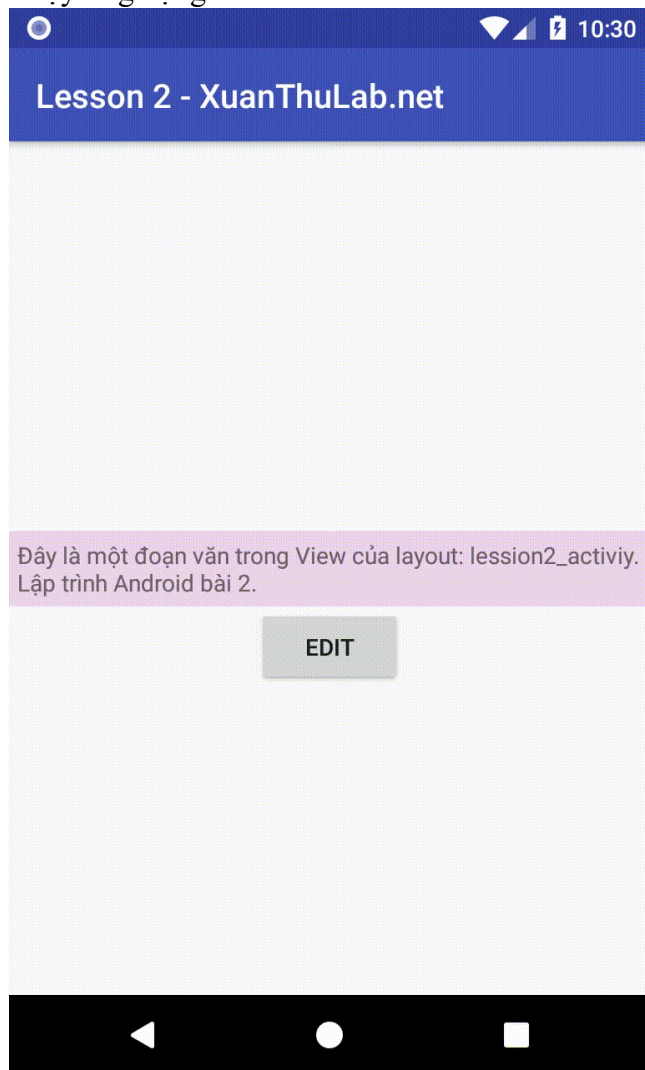
```

```
//Nếu có truyền tham số cho Activity
intent.putExtra("noidung", textView.getText());

//Gửi Intent cho hệ thống Android để kích hoạt Activity
startActivity(intent);
//Muốn Activity thứ nhất kết thúc thì thêm finish();

    }
};
//Gán Listener cho editbutton
edibutton.setOnClickListener(listenerEditbutton);
```

Chạy ứng dụng:



Sử dụng Intent chứa kết quả trả về

Trong `UpdateActivity` nếu muốn nó trả kết quả về thì trước khi `finish()`; cần tạo một đối tượng **Intent**, rồi chuyển dữ liệu cần trả về vào đó như cách ở trên, cuối cùng gọi phương thức `setResult(code_result, data)`;

Bằng cách như vậy thì ở Activity khác có thể nhận được dữ liệu trả về từ `UpdateActivity` (chú ý các code trên, mặc dù `UpdateActivity` có trả về dữ liệu thì `Lesson2Activity` vẫn chưa nhận được dữ liệu, ta cần chỉnh sửa một chút)

Áp dụng cập nhật code cho `UpdateActivity` sẽ trả về dữ liệu có chứa dòng Text mới cập nhật lại.

```
package net.xuanthulab.firstapp;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

/**
 * Created by XUANTHULAB.NET on 15-Mar-18.
 */

public class UpdateActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.update_activity);

        EditText noidung_edit = findViewById(R.id.noidung_edit);

        Intent intent = getIntent();
        String noidung = intent.getStringExtra("noidung");
        noidung_edit.setText(noidung);

        Button ok_button = findViewById(R.id.ok_button);
        ok_button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                setResultData();
                finish();
                Toast.makeText(view.getContext(), "Save", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void setResultData()
    {
        Intent data = new Intent();
        EditText editText = findViewById(R.id.noidung_edit);
        String noidung = editText.getText().toString();
        data.putExtra("noidung", noidung);

        int code_result = 1;
        setResult(code_result, data);
    }
}
```

```
}  
  
}
```

Nhận kết quả trả về từ một Activity khác

Ở phần trước, chúng ta sử dụng phương thức **startActivity** để khởi động Activity, nếu vậy thì nó không nhận được kết quả trả về. Nếu muốn nhận kết quả trả về thì khởi động Activity bằng phương thức **startActivityForResult(Intent intent, int requestCode)**, trong đó requestCode là một số nguyên để bạn kiểm tra xem kết quả nhận về từ Activity nào.

Nếu gọi như trên, thì Activity sẽ nhận dữ liệu thông qua phương thức **onActivityResult**, bạn chỉ việc nạp chồng hàm này và lấy dữ liệu trả về.

Cập nhật lại code **Lesson2Activity.java** để nhận dữ liệu trả về và cập nhật lại vào TextView

```
public class Lesson2Activity extends AppCompatActivity {  
    final int RESULT_UPDATE_ACTIVITY = 1;  
    @Override  
    protected void onCreate(@Nullable Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.lesson2_activity);  
  
        final TextView textView = findViewById(R.id.noidung);  
  
        Button edibutton = findViewById(R.id.editbutton);  
        //Tạo listener cho sự kiện Click  
        View.OnClickListener listenerEditbutton = new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
  
                Intent intent = new Intent();  
  
                //Gán context và tên lớp Activity cần chạy  
                intent.setClass(view.getContext(), UpdateActivity.class);  
  
                //Nếu có truyền tham số cho Activity  
                intent.putExtra("noidung", textView.getText());  
  
                //Gửi Intent cho hệ thống Android để kích hoạt Activity  
                startActivityForResult(intent, RESULT_UPDATE_ACTIVITY);  
                //Muốn Activity thứ nhất kết thúc thì thêm finish();  
  
            }  
        };  
        //Gán Listener cho editbutton  
        edibutton.setOnClickListener(listenerEditbutton);  
    }  
}
```

```
}

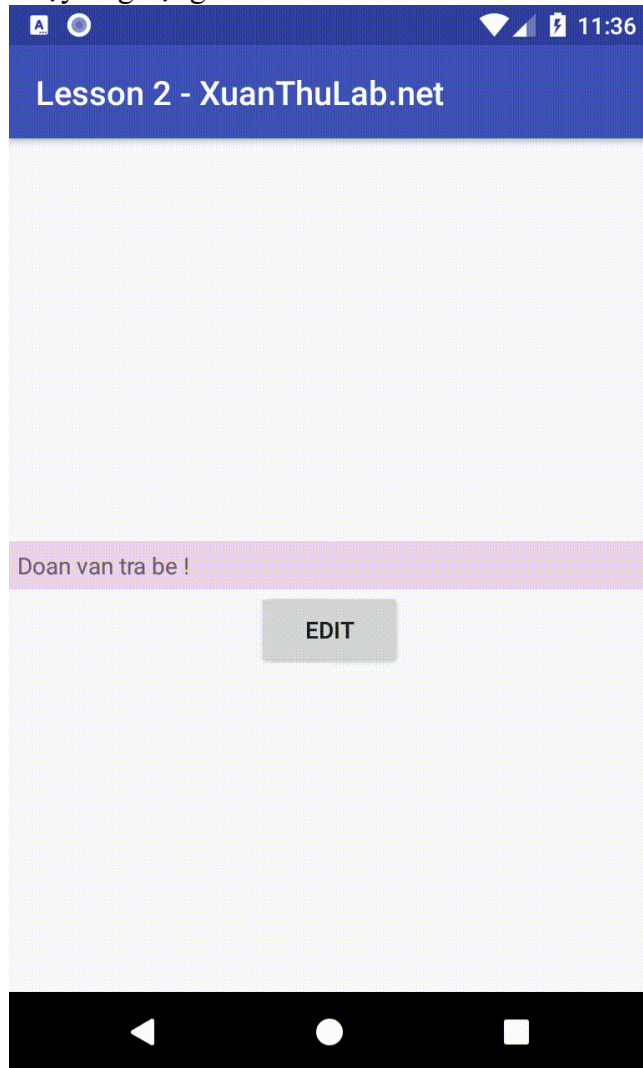
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    switch (requestCode) {
        case RESULT_UPDATE_ACTIVITY:
            String noidung = data.getStringExtra("noidung");
            TextView textView = findViewById(R.id.noidung);
            textView.setText(noidung);
            break;
            default: break;
    }
}

@Override
protected void onPause() {
    super.onPause();

    //Code to save data
}
}
```

Chạy ứng dụng:



Ta thấy kết quả trả về từ Activity thứ 2 nhận được bởi Activity thứ nhất.

Mục lục bài viết

[Cơ bản về Activity](#)[Sử dụng Intent để mở một Activity](#)[Đọc dữ liệu truyền đến bằng Intent](#)[Dùng Intent để trả kết quả về](#)[Đọc kết quả trả về](#)

AJ-03: Tổng quan chạy, trao đ

Đăng ký theo dõi ủng hộ kênh



XuanThuLab

YouTube

Thích 2

Chia sẻ 2



0 bình luận

Sắp xếp theo **Mới nhất**



Thêm bình luận...

Plugin bình luận trên Facebook

[SQLite trong Android \(phần 2\)](#) [sử dụng SQLiteOpenHelper](#) [Sử dụng SQLite trong Android \(phần 1\)](#) [Sử dụng ListView hiển thị dữ liệu dạng danh sách trong Android](#) [Tạo và sử dụng Theme trong lập trình Android](#) [Thuộc tính style trong các View lập trình Android](#) [Hiểu về View, ViewGroup, Layout trong Android](#) [Lưu và phục hồi trạng thái của Activity trong Android](#) [Các thành phần của một ứng dụng Android](#) [Cài đặt Android Studio, máy ảo Android, tạo dự án ứng dụng đầu tiên](#) [CoordinatorLayout \(2\)](#) [ứng dụng một số loại View](#)

[Vòng đời Activity](#)

[Lưu và phục hồi trạng thái](#)

[Giới thiệu PrivacyPolicy](#) [Liên hệ RSS](#)

Đây là blog cá nhân, tôi ghi chép và chia sẻ những gì tôi học được ở đây về kiến thức lập trình PHP, Java, JavaScript, Android, C# ... và các kiến thức công nghệ khác

Developed by [XuanThuLab](#)

DMCA PROTECTED

