

- Học 15 tuần (2/02/2023 – 11/05/2023)
- Nội dung: được cập nhật trên utex và fhq
- Đánh giá: 50% quá trình (các bài tập trên lớp và bài tập theo project cuối khóa), 50% bài project cuối khóa.
- Liên hệ: Nguyễn Hữu Trung
- Khoa CNTT – DHSPKT
- Email: [trungnh@hcmute.edu.vn](mailto:trungnh@hcmute.edu.vn)
- Sdt: 0908 617108

- Viết app bán hàng....
- Viết app mạng xã hội...
- Viết app tư vấn sinh viên
- Viết app phục vụ nhu cầu của trường sư phạm kỹ thuật



KHOA CÔNG NGHỆ THÔNG TIN



## GIỚI THIỆU ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



ThS. Nguyễn Hữu Trung

- Android là một hệ điều hành mã nguồn mở và là một hệ điều hành dựa trên Linux. Ban đầu Android được phát triển bởi Công ty Android với sự hỗ trợ tài chính từ Google, sau đó được Google mua lại vào năm 2005.
- Phiên bản beta của Android Software Development Kit (SDK) được công bố bởi Google vào năm 2007, trong khi phiên bản thương mại đầu tiên Android 1.0 được công bố 9/2008.
- Vào 27/6/2012, tại hội nghị Google I/O, Google công bố phiên bản Android tiếp theo là 4.1 Jelly Bean. Jelly Bean là một bản cập nhật với mục đích đầu tiên là cải thiện giao diện người dùng (User Interface), cả về tính năng lẫn hiệu suất.
- Mã nguồn cho Android là miễn phí. Google công bố hầu hết các code dưới Apache License version 2.0, và phần còn lại, các thay đổi Linux Kernel dưới GNU General Public License version 2.

- **Android** là mã nguồn mở
- Bất kỳ ai cũng tùy biến lại **Android Platform**
- Người sử dụng có thể lựa chọn nhiều ứng dụng trên **Android**.
- **Android** cung cấp nhiều tính năng thú vị như: thời tiết, mở màn hình, RSS (Really Simple Syndication) vv
- **Android** cũng hỗ trợ cho các dịch vụ nhắn tin (SMS và MMS), trình duyệt web, lưu trữ dữ liệu (SQLite), kết nối (GSM, CDMA, BlueTooth, Wi-Fi vv), media....

- Android chiếm hơn 87,7%(2017), 69.74% (2022) thị phần điện thoại thông minh trên toàn thế giới với khoảng hơn 2 tỷ thiết bị đã được kích hoạt và hơn 1,3 triệu lượt kích hoạt mỗi ngày.



Music



News



Multimedia



Sports



Lifestyle



Food & Drink



Travel



Weather



Books



Business



Reference



Navigation



Social Media



Utilities



Finance

- Hiện tại, dãy Codename của Android từ A tới T, như Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat và Lollipop,..., Pie 2018, Android 10 Q (2019), Red velvet cake(2020), Snow Cone(2021), Tiramisu (2022).





KHOA CÔNG NGHỆ THÔNG TIN



## CÀI ĐẶT JDK và ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



ThS. Nguyễn Hữu Trung



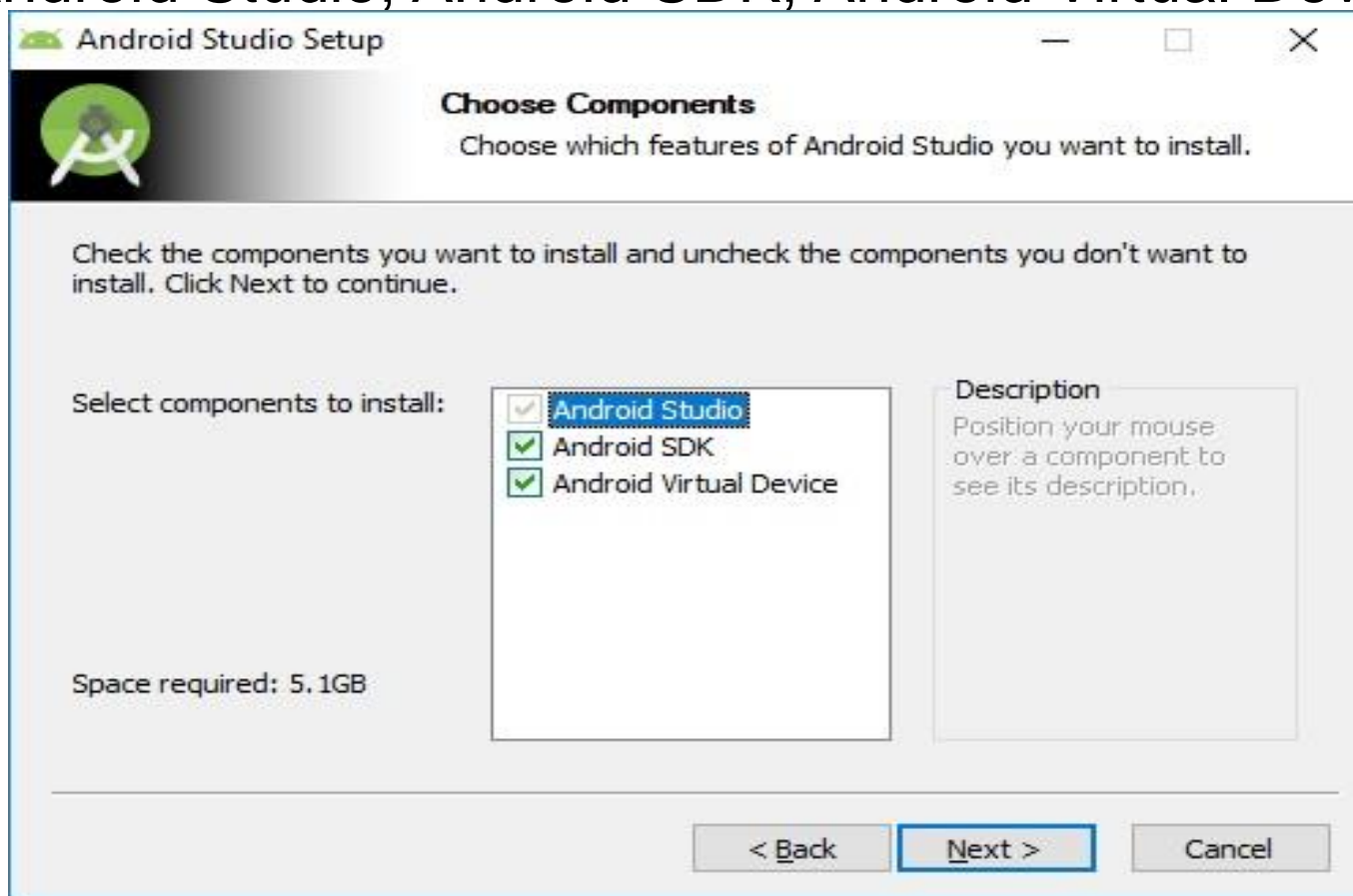
- Tải JDK 8 trở lên và cài đặt vào máy.
- Cấu hình môi trường JDK trên máy tính.

- Bạn có thể phát triển ứng dụng [Android](#) ở môi trường Windows, Mac OS và cả Linux. Ngoài ra có rất nhiều công cụ để xây dựng các ứng dụng Android phải kể đến như Android Studio, IntelliJ IDEA, Eclipse, Xamarin...
  - ▣ Thiết lập JDK (Java Development Kit)
  - ▣ Thiết lập Android IDEs

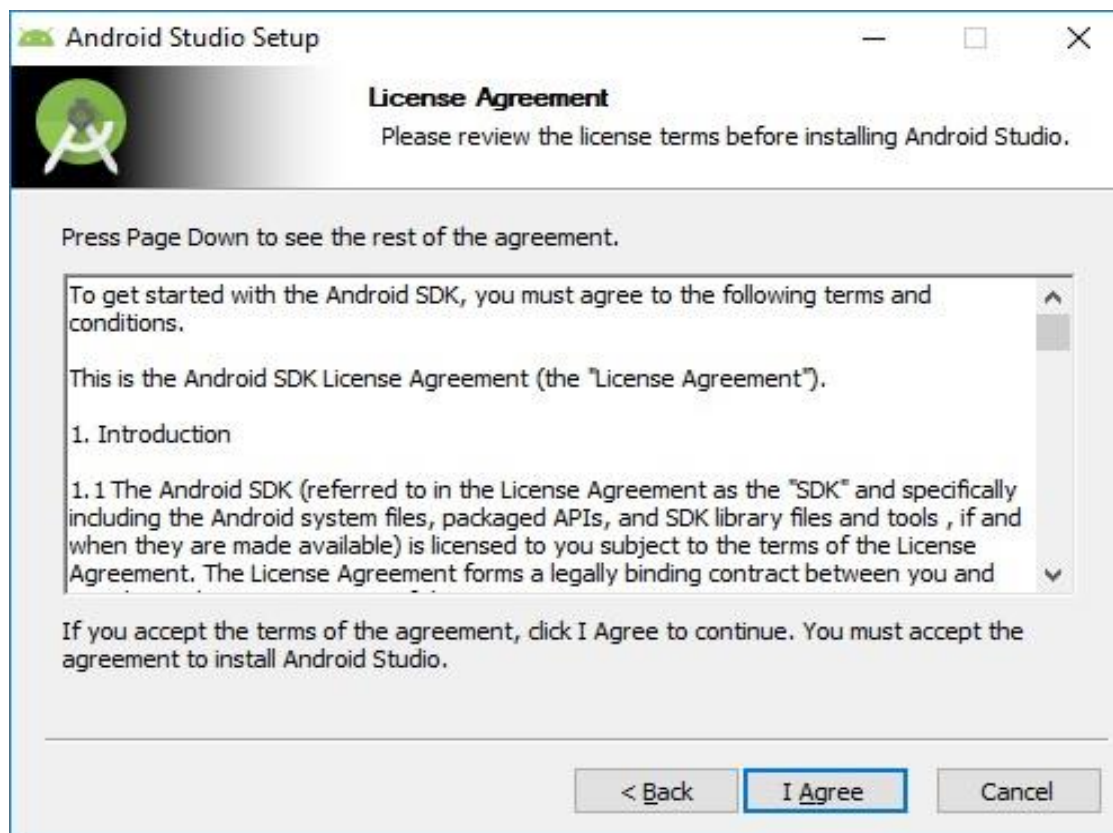
- **Bước 1:** Bạn nhấp đúp chuột vào tập tin Android Studio vừa tải về, sau đó nhấn Next để xác nhận cài đặt Android Studio.



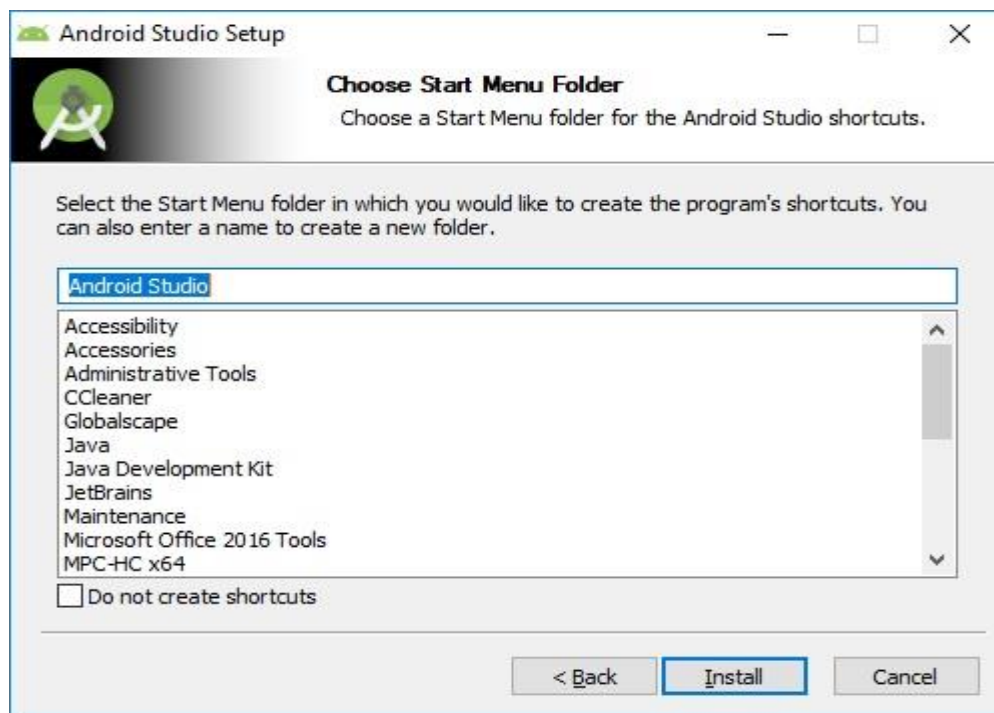
- Bạn hãy tích chọn tất cả các tùy chọn hiện có bao gồm Android Studio, Android SDK, Android Virtual Device.



- Đọc xong phần giấy phép, điều khoản xong thì nhấn chọn **I Agree** để đồng ý cài đặt Android Studio.



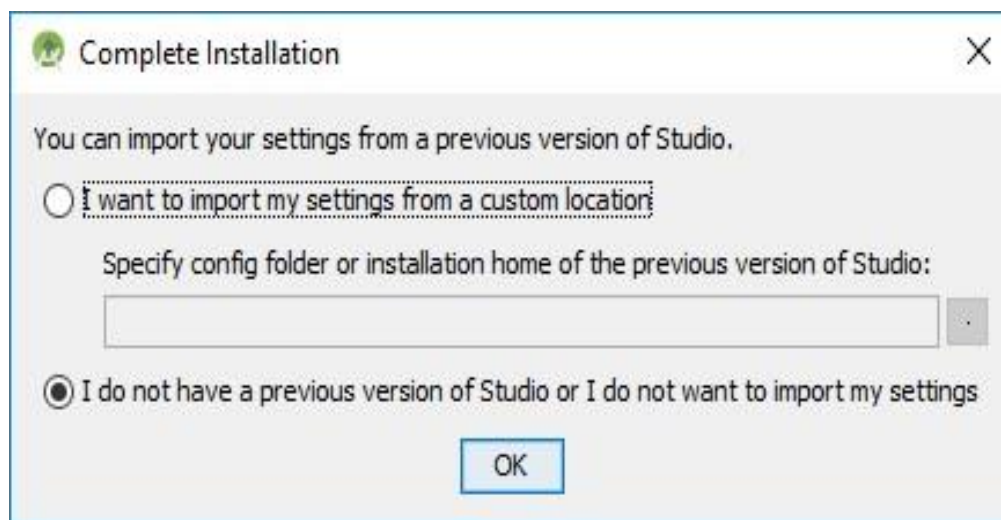
- Màn hình tiếp theo là lựa chọn nơi lưu trữ, bạn có thể để mặc định và nhấn **Next** để qua bước kế tiếp.
- Bây giờ hãy nhấn chọn **Install** để bắt đầu tiến hành cài đặt Android Studio.



- Hãy chờ đợi quá trình cài đặt diễn ra tùy cấu hình từng máy, sau có thông báo cài đặt thành công bạn nhấn **Finish**.

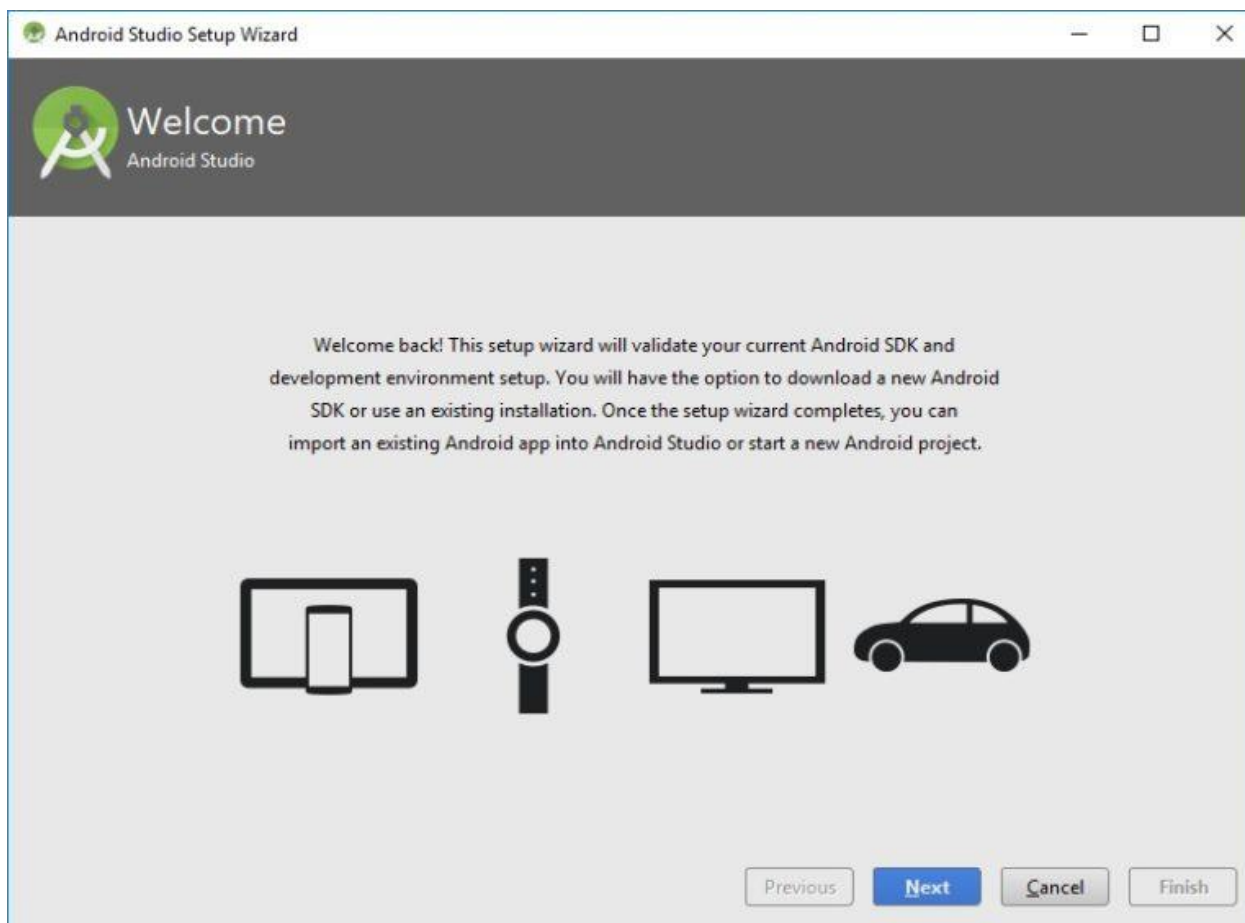


- **Bước 2:** Bây giờ bạn hãy nhấn chọn các bước như sau đây.

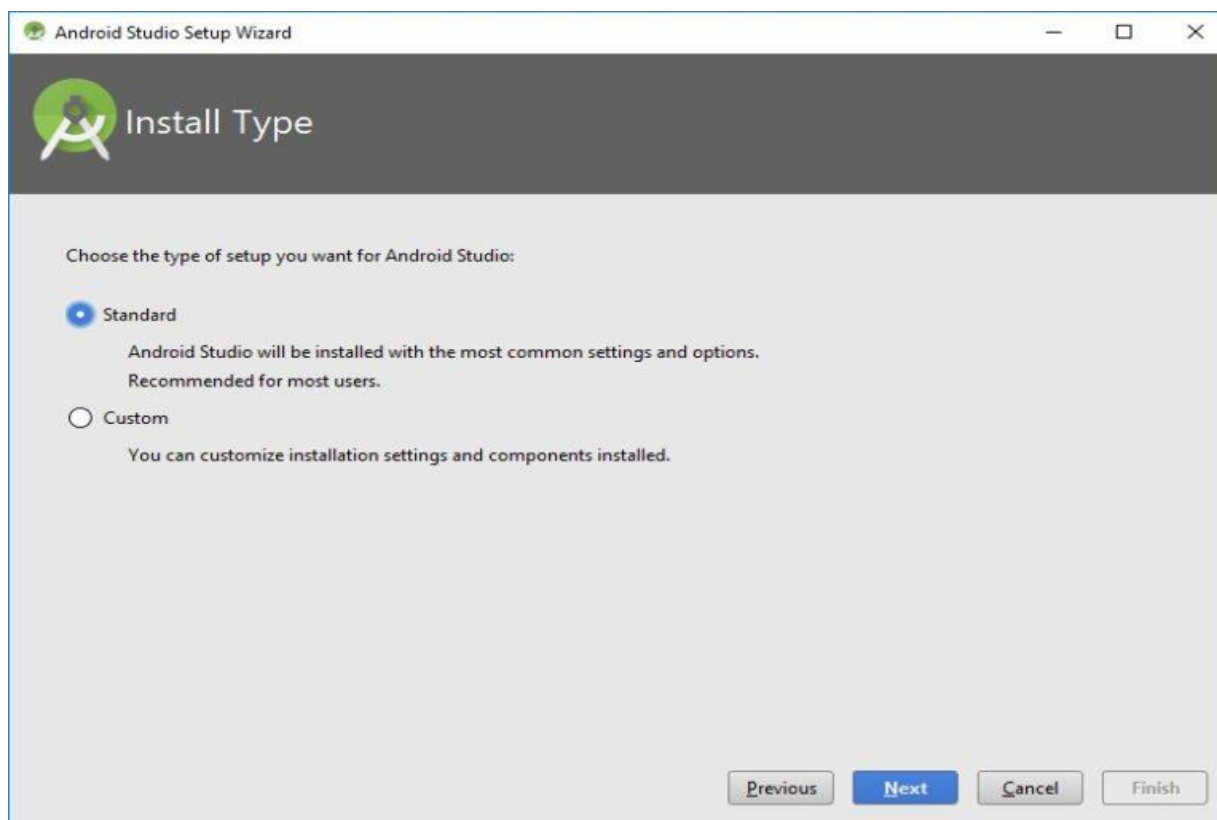




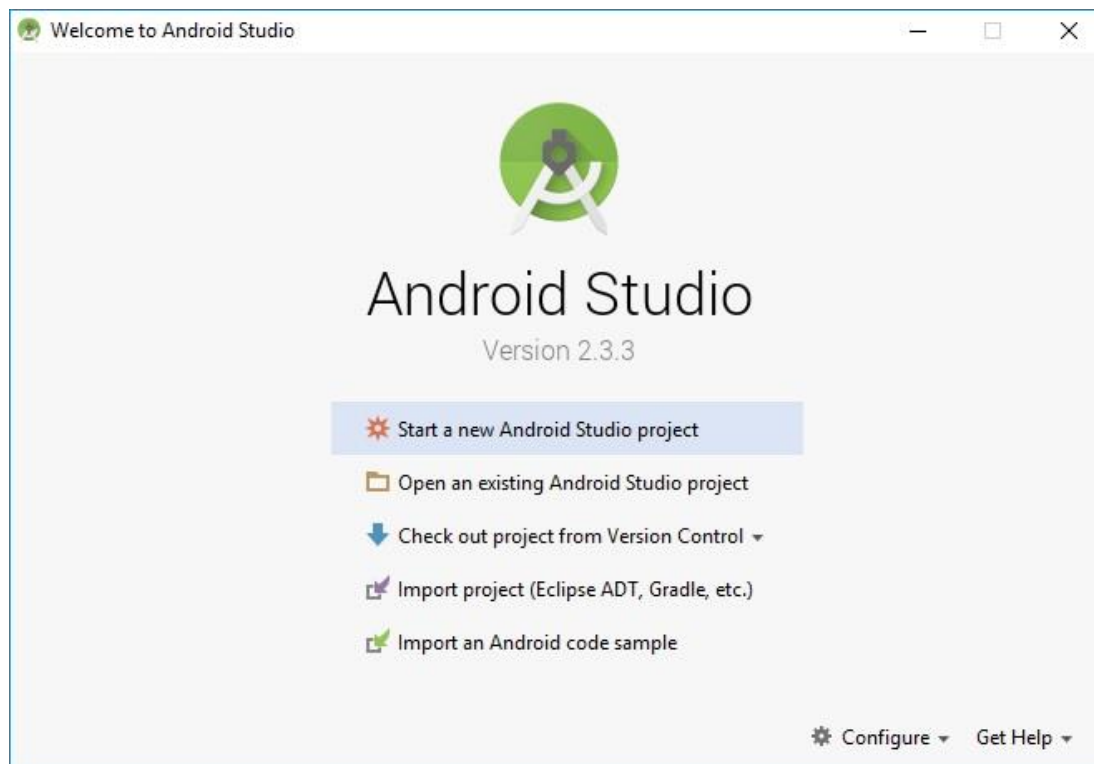
- Nhấn **Next** để qua bước kế tiếp.



- Nhấn **Next** để qua bước kế tiếp. Rồi nhấn **Finish** để qua bước kế tiếp



- **Bước 3:** Tiếp tục chờ đợi để quá trình cài đặt diễn ra, sau khi cài đặt xong nhấn **Finish** để hoàn thành quá trình.





KHOA CÔNG NGHỆ THÔNG TIN



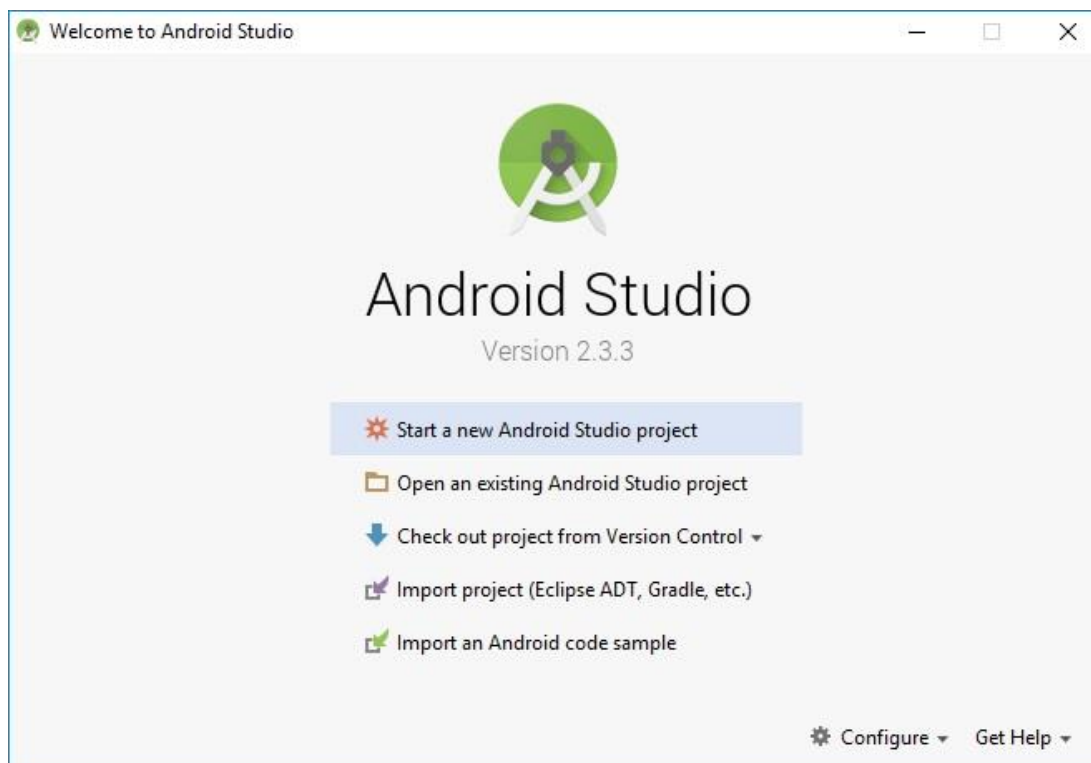
## TẠO PROJECT ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

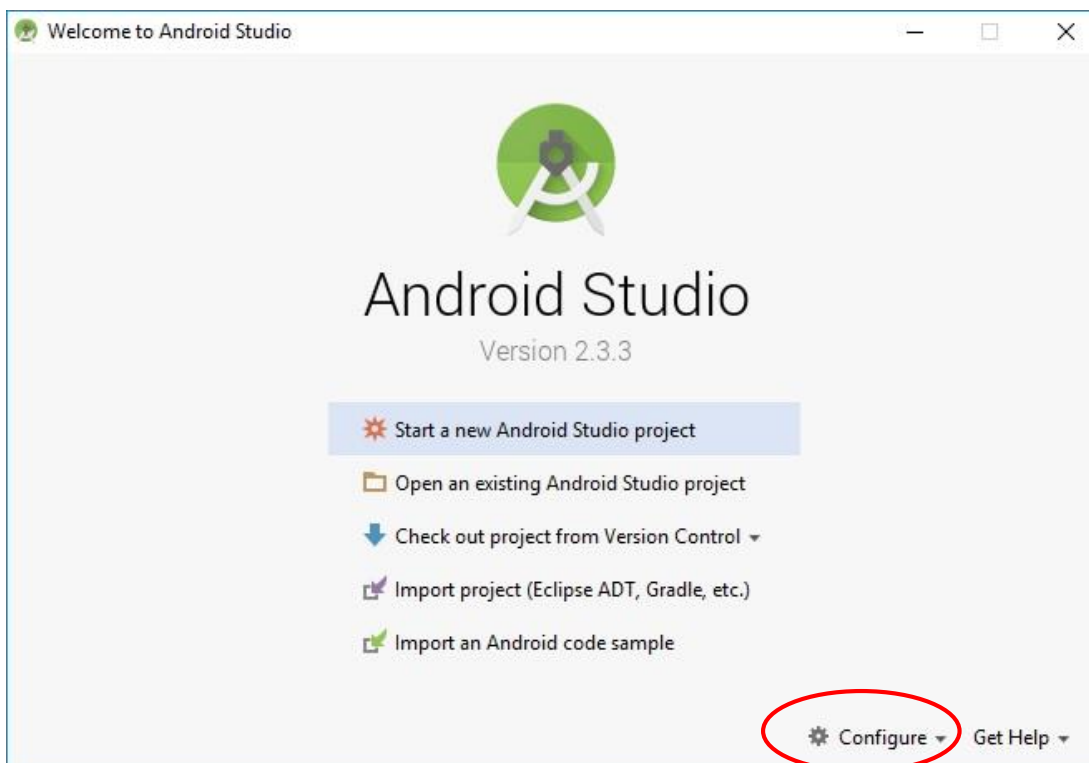


ThS. Nguyễn Hữu Trung

- Khởi động Android Studio lần đầu tiên, bạn chọn **Configure** để cấu hình máy ảo mô phỏng.



- Khởi động Android Studio lần đầu tiên, bạn chọn Configure để cấu hình máy ảo mô phỏng (Virtual Device manager).



**Device Manager**

Create device ?

Device ▲

- Pixel XL API 33  
Android Tiramisu Google APIs | x86\_64
- Pixel\_3a\_API\_33\_x86\_64  
Android Tiramisu Google APIs | x86\_64

**Virtual Device Configuration**

Select Hardware

Choose a device definition

Category	Name ▼	Play Store	Size	Resolution	Density
Phone	Resizable (Experiment...		6.0"	1080x2340	420dpi
Tablet	Pixel XL		5.5"	1440x2560	560dpi
Wear OS	Pixel 6 Pro		6.7"	1440x3120	560dpi
TV	Pixel 6		6.4"	1080x2400	420dpi
Automotive	Pixel 5		6.0"	1080x2340	440dpi
	Pixel 4a		5.8"	1080x2340	440dpi
	Pixel 4 XL		6.3"	1440x3040	560dpi

Pixel 2

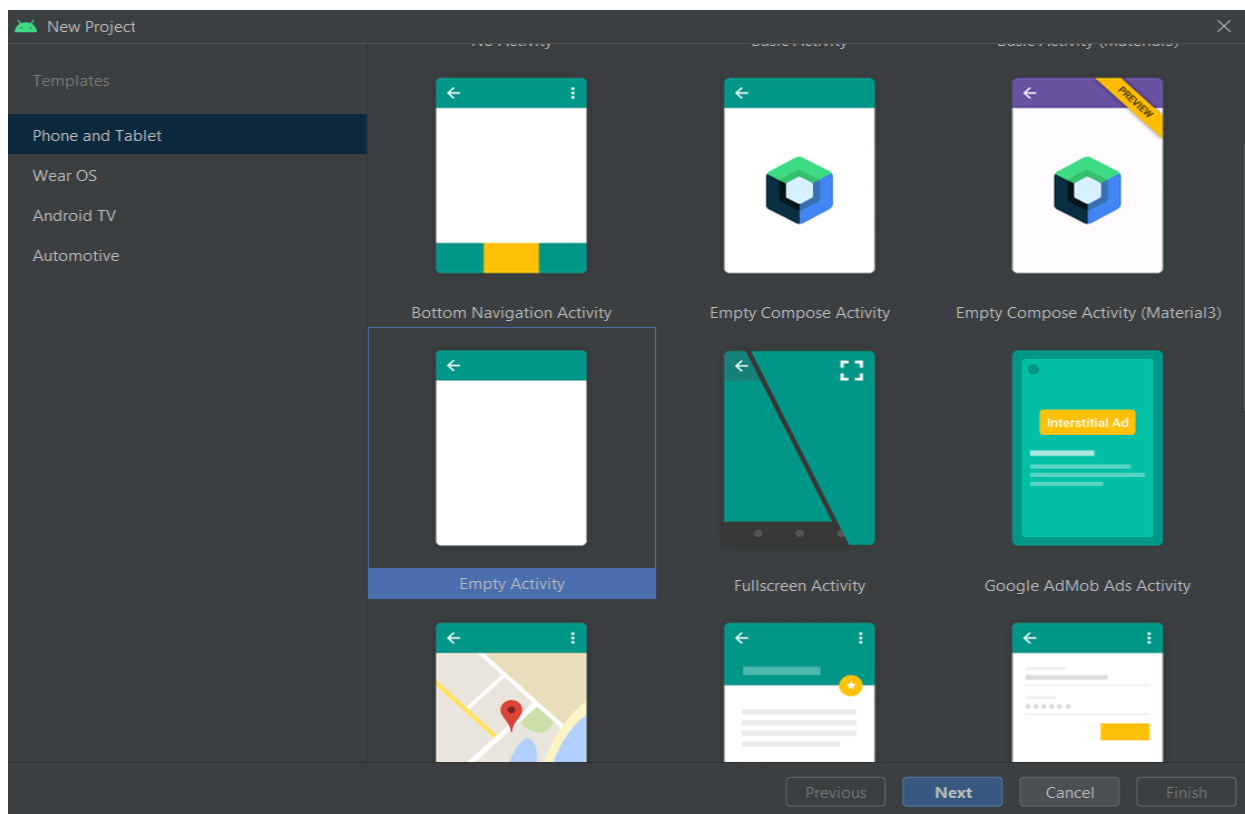
Size: large  
Ratio: long  
Density: 420dpi

1080px  
5.0"  
1920px

New Hardware Profile Import Hardware Profiles Clone Device...

Previous Next Cancel Finish

- Khởi động Android Studio chọn New project rồi chọn Empty Activity





New Project

**Empty Activity**

Creates a new empty activity

Name: My Application

Package name: vn.iotstar.myapplication

Save location: C:\Users\trungnh\AndroidStudioProjects\MyApplication2

Language: Java

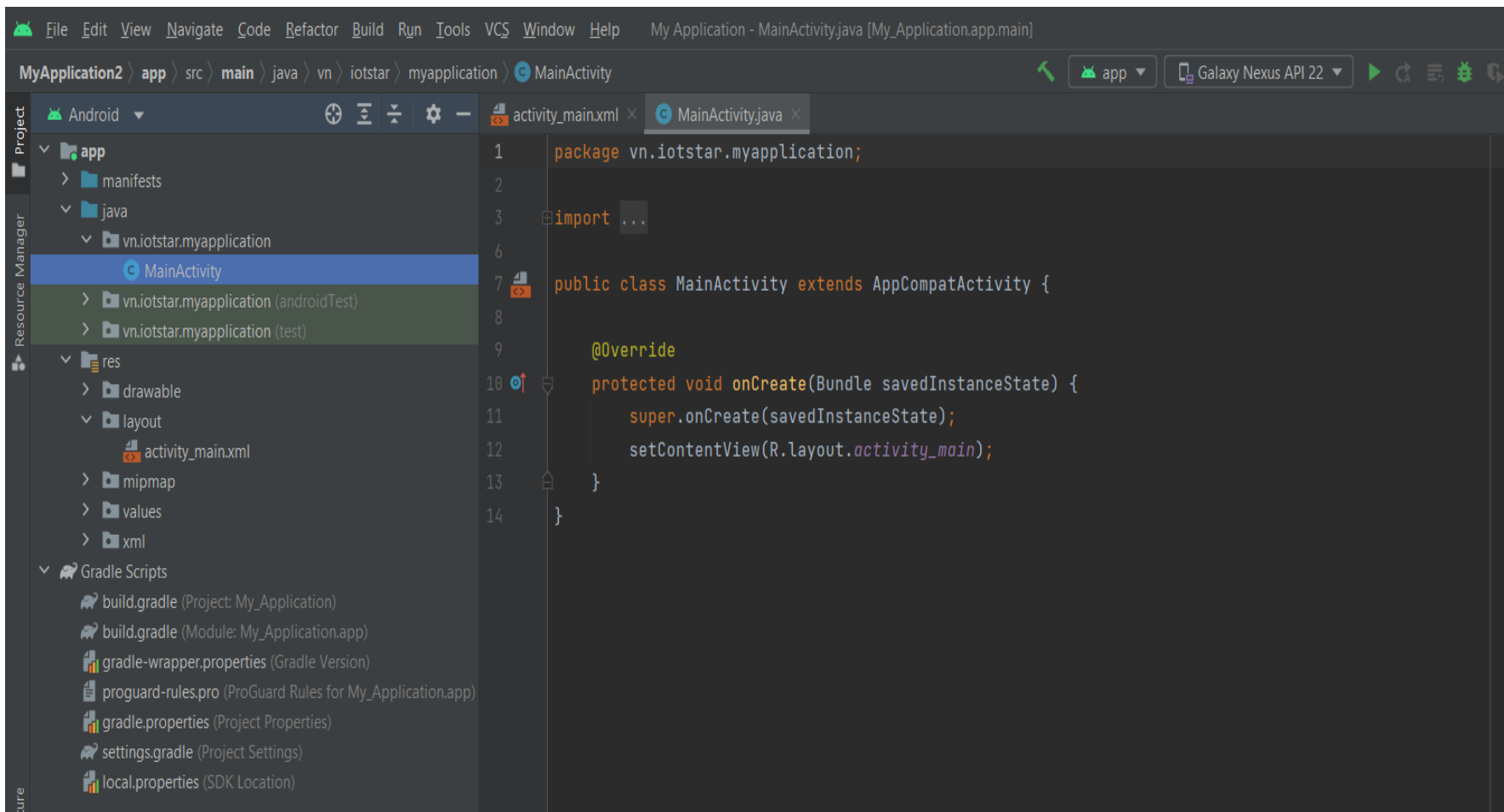
Minimum SDK: API 16: Android 4.1 (Jelly Bean)

**i** Your app will run on approximately **100%** of devices.  
[Help me choose](#)

☐ Use legacy android.support libraries **?**  
 Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

Previous Next Cancel **Finish**

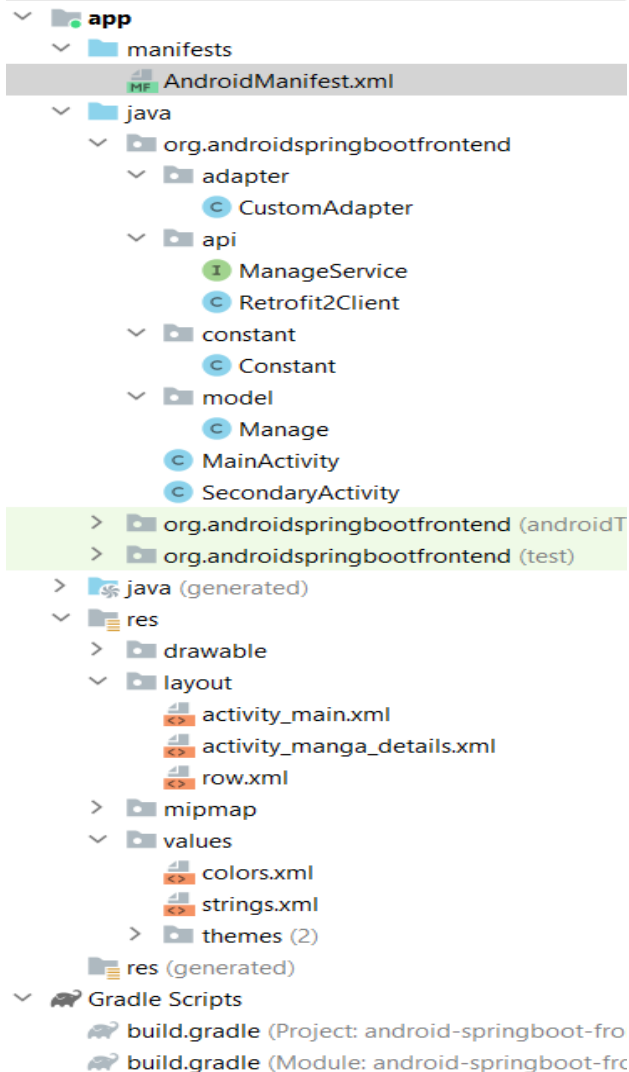
- Chọn các thông tin rồi bấm Finish



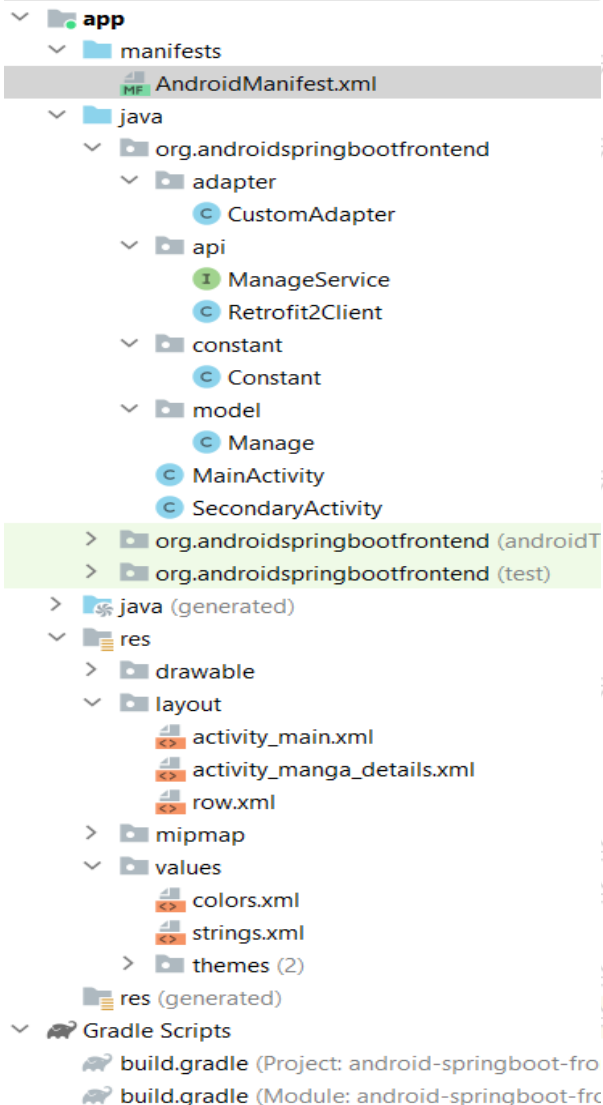
```

1  package vn.iotstar.myapplication;
2
3  import ...
6
7  public class MainActivity extends AppCompatActivity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14 }

```

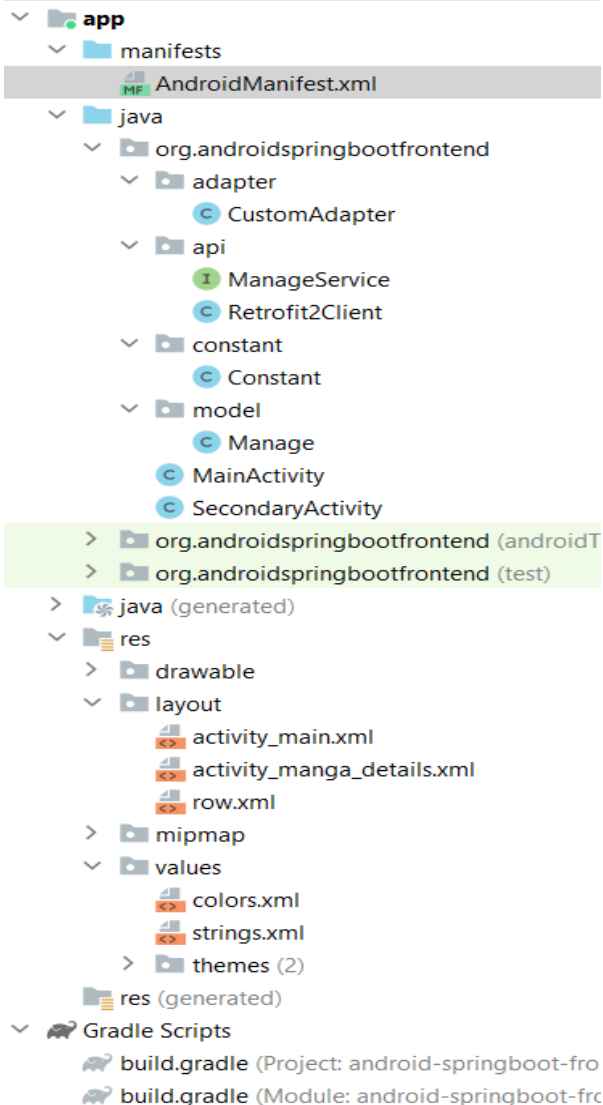


- **Thư mục manifests:** chứa thông tin cấu hình của ứng dụng. **AndroidManifest.xml:** tập tin XML chứa tất cả các thông tin cấu hình dùng để build ứng dụng và các thành phần của ứng dụng (activity, service, broadcast receivers, content providers v.v.). Mỗi ứng dụng đều có một tập tin AndroidManifest.xml. Trong ứng dụng, Activity nào muốn sử dụng đều bắt buộc phải có khai báo trong AndroidManifest.xml

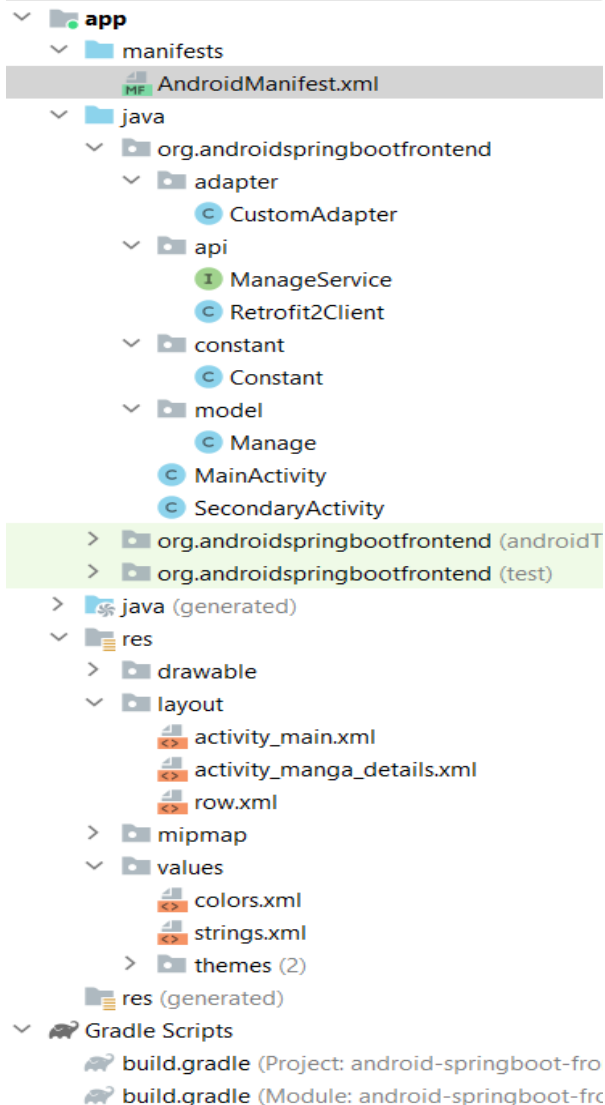


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="android-springboot-frontend"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Androidspringbootfrontend"
        android:usesCleartextTraffic="true">
        <activity
            android:name=".MainActivity"
            android:label="android-springboot-frontend"
            android:theme="@style/Theme.AppCompat.NoActionBar"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SecondaryActivity" />
    </application>
</manifest>
```

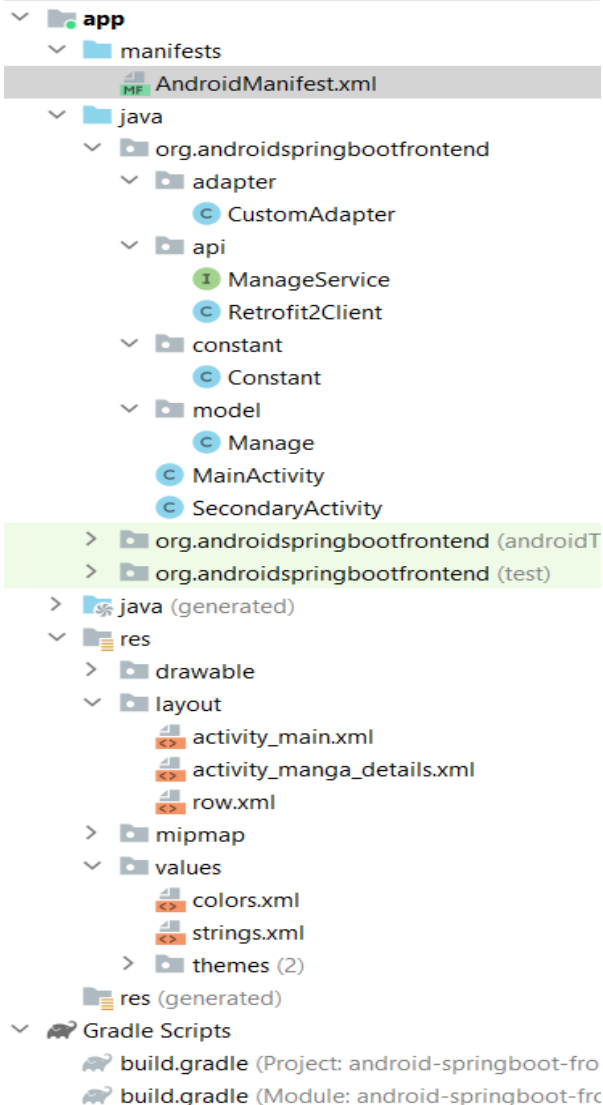


- **Thư mục java:** chứa tất cả các file mã nguồn .java của ứng dụng. Tương ứng với mỗi **Activity** thì file mã nguồn sẽ chứa các xử lý trên Activity đó. Activity nào được khởi chạy đầu tiên khi ứng dụng hoạt động sẽ được khai báo đầu tiên trong tập tin **AndroidManifest.xml**.
- **Thư mục res:** chứa các tài nguyên của ứng dụng, bao gồm các tập tin hình ảnh, các thiết kế giao diện, thực đơn,... của ứng dụng.



- **Thư mục res:** chứa các tài nguyên của ứng dụng, bao gồm các tập tin hình ảnh, các thiết kế giao diện, thực đơn,... của ứng dụng.

Thư mục	Mô tả
<b>drawable/</b>	Thư mục chứa các hình ảnh của ứng dụng
<b>layout/</b>	Thư mục chứa các tập tin XML dùng để thiết kế giao diện của ứng dụng. Hiện giờ đang có tập tin <i>activity_main.xml</i> chính là phần thiết kế giao diện của ứng dụng <i>Hello world</i> .
<b>mipmap/</b>	Thư mục thường chứa các hình ảnh dạng icon của ứng dụng.
<b>values/</b>	Thư mục quản lý các giá trị, đó có thể là các giá trị chuỗi (strings.xml), màu sắc (colors.xml),...

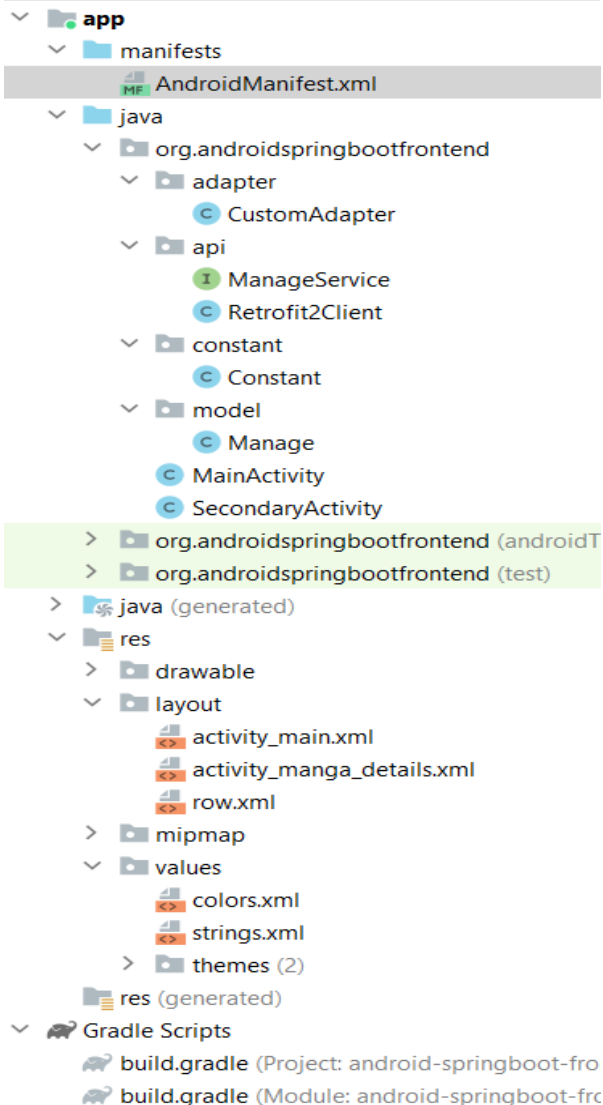


## Thiết lập và truy xuất Android Resources

```
res/
  drawable/
    icon.png
  layout/
    activity_main.xml
    info.xml
  values/
    strings.xml
```

Thư mục **res/** chứa tất cả các thư mục con. Trong ví dụ trên chúng ta có **1 image resource**, **2 layout resource**, và **1 string resource**.

- **anim/**: Chứa các file xml định nghĩa các hiệu ứng. Chúng được truy xuất từ class **R.anim**.
- **color/**: Là các file xml định nghĩa về màu sắc. Chúng được truy xuất từ class **R.color**.
- **drawable/**: Chứa các file hình ảnh **.png**, **.jpg**, **.gif** hoặc các file xml đã được biên dịch thành bitmaps. Chúng được truy xuất từ class **R.drawable**.
- **layout/**: Chứa các file xml định nghĩa giao diện người dùng. Chúng được truy xuất từ class **R.layout**.



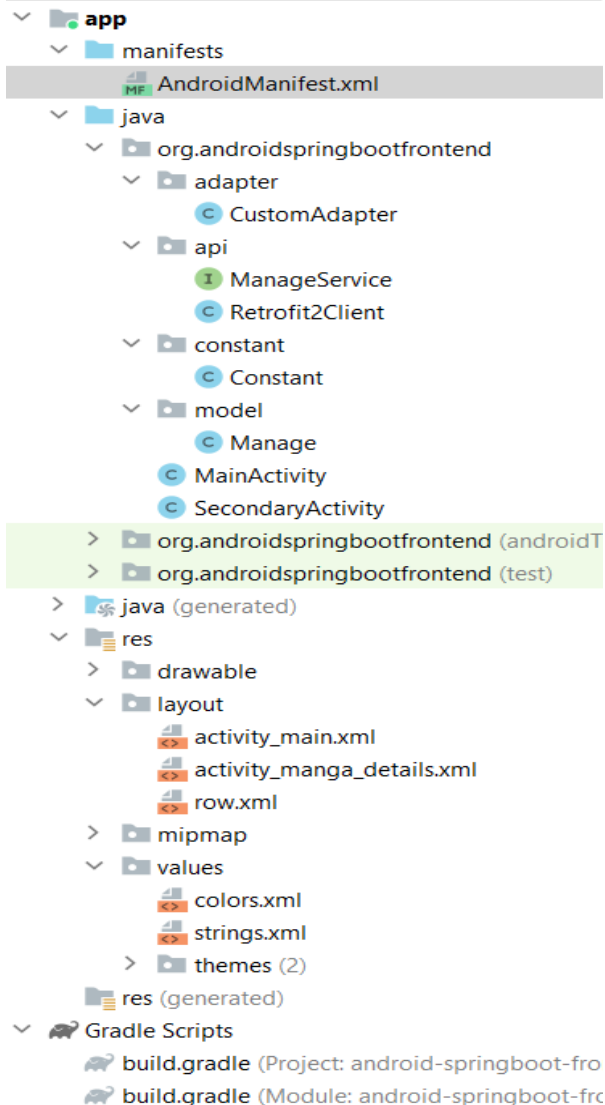
## Thiết lập và truy xuất Android Resources

```
res/
  drawable/
    icon.png
  layout/
    activity_main.xml
    info.xml
  values/
    strings.xml
```

Thư mục **res/** chứa tất cả các thư mục con. Trong ví dụ trên chúng ta có **1 image resource**, **2 layout resource**, và **1 string resource**.

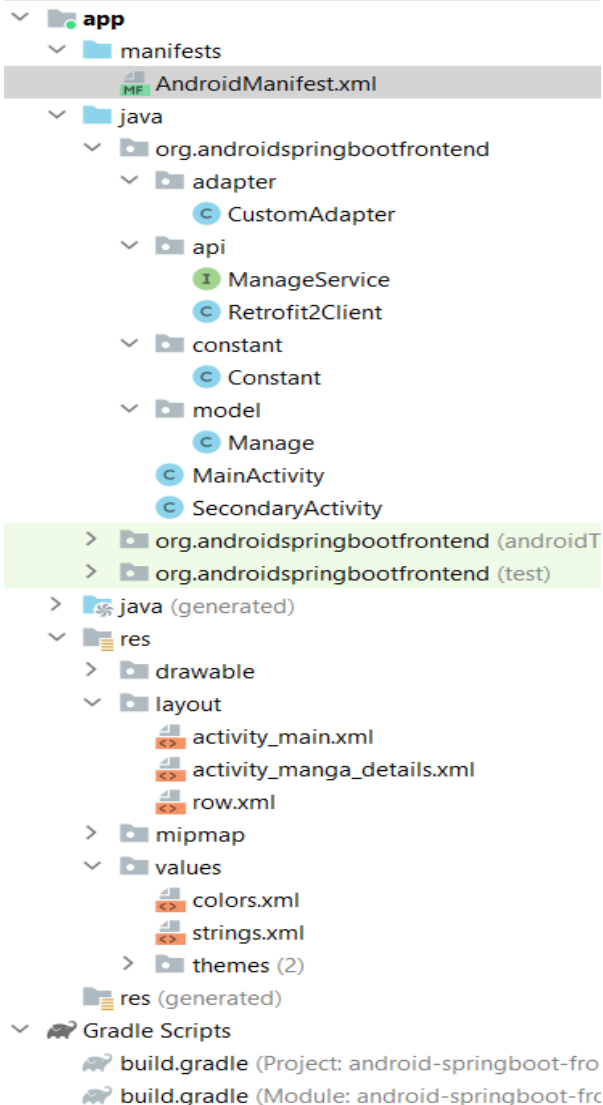
- **menu/**: Chứa các file xml định nghĩa menu của ứng dụng, có thể là Options Menu, Context Menu, Sub Menu. Chúng được truy xuất từ class **R.menu**.
- **raw/**: Chứa các file bất kỳ, tùy ý. Cần phải gọi **Resources.openRawResource()**. Với **resource ID**, nó là **R.raw.filename** để mở 1 file **raw**.





## Thiết lập và truy xuất Android Resources

- **values/**: Chứa các file xml để lưu các giá trị đơn giản như string, integers, colors, ...
  - + **arrays.xml**: Dùng để định nghĩa các mảng dữ liệu, và được truy xuất từ class **R.array**.
  - + **integers.xml**: Dùng để định nghĩa các số integer và được truy xuất từ class **R.integer**.
  - + **bools.xml**: Dùng để định nghĩa kiểu dữ liệu bool và được truy xuất từ class **R.bool**.
  - + **colors.xml**: Dùng để định nghĩa các giá trị color và được truy xuất từ class **R.color**.



## Thiết lập và truy xuất Android Resources

- **values/**: Chứa các file xml để lưu các giá trị đơn giản như string, integers, colors, ...
  - + **dimens.xml**: Dùng để định nghĩa các giá trị về kích thước, chiều. Được truy xuất từ class **R.dimen**.
  - + **strings.xml**: Dùng để định nghĩa các string và được gọi từ class **R.string**.
- **styles.xml**: Dùng để định nghĩa các styles và được gọi từ class **R.style**.
- **xml/**: Chứa các file xml bất kỳ và có thể được đọc bằng cách gọi hàm `Resources.getXML()`. Bạn có thể lưu các file XML cấu hình ở đây và nó sẽ được dùng khi ứng dụng chạy.

## Thiết lập và truy xuất Android Resources

**Ví dụ:** Truy xuất **res/drawable/myImage.png**

```
ImageView imageView = (ImageView)
findViewById(R.id.myimageView);
```

```
imageView.setImageResource(R.drawable.myimage);
```

**Ví dụ:** Truy xuất **res/values/strings.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello, World!</string>
</resources>
```

Để sử dụng string: **'hello'**

```
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello);
```

## Truy xuất resources trong XML

**Ví dụ:** Truy xuất **res/values/strings.xml**

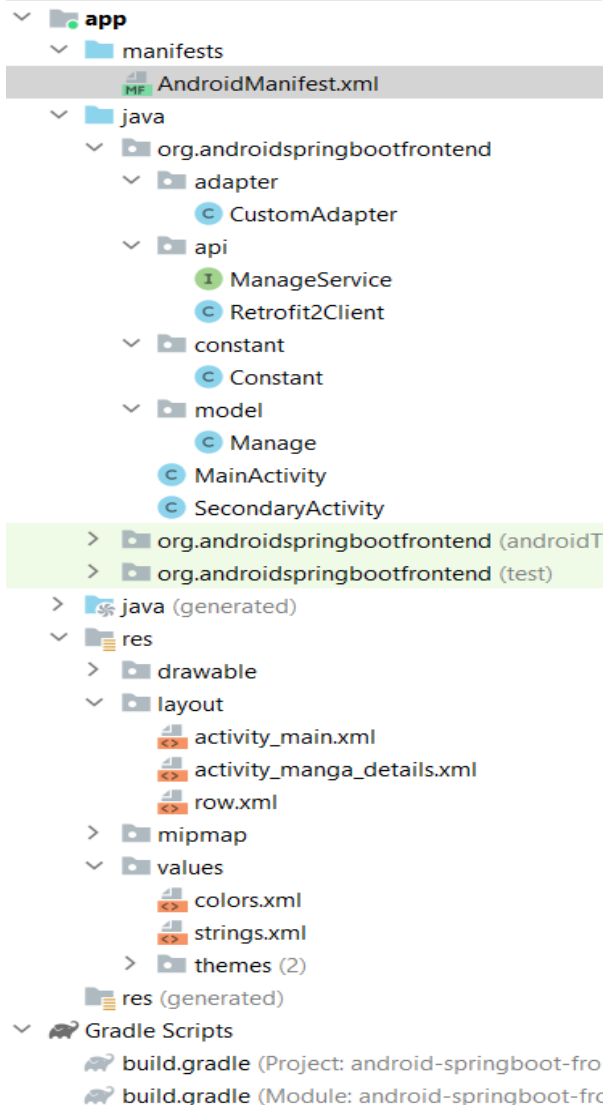
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <color name="opaque_red">#f00</color>

    <string name="hello">Hello!</string>

</resources>
```

```
android:textColor="@color
/opaque_red"
android:text="@string/hel
lo" />
```

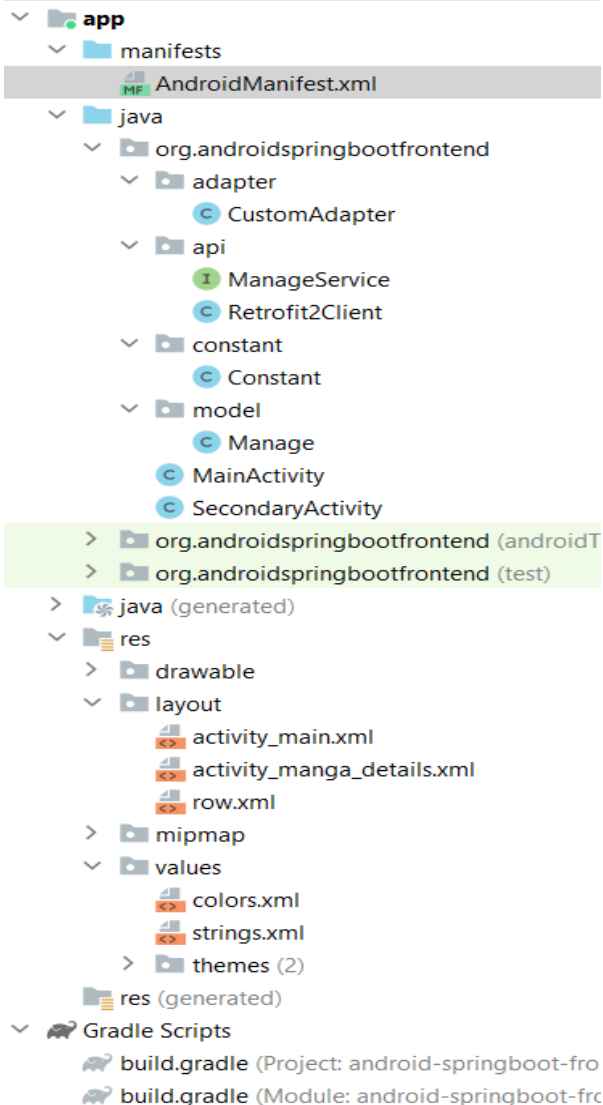


**Android** R.java là một tập tin được tạo tự động bởi aapt (Android Asset Packaging Tool) .Nó chứa tất cả các ID trong thư mục res /. Nó chứa nhiều class tĩnh : **menu, id, layout, attr, drawable, string .v.v**

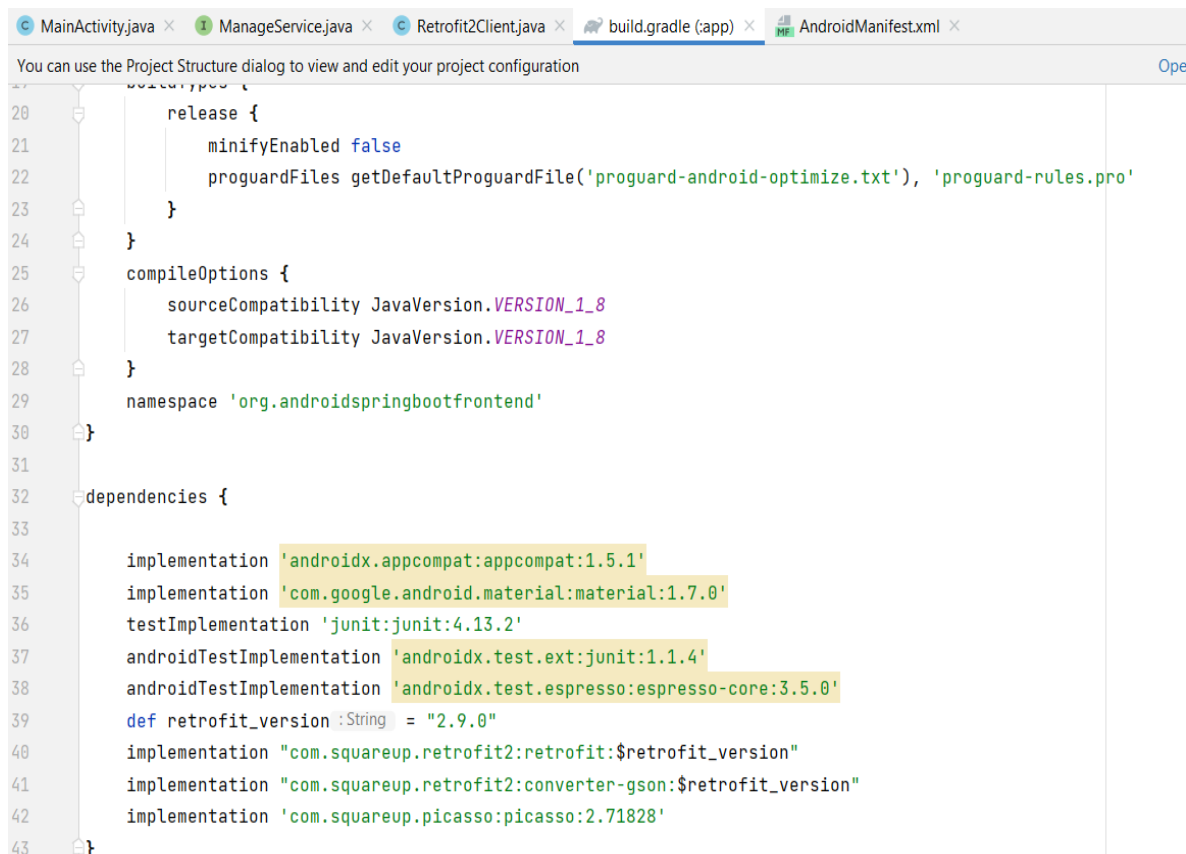
Truy xuất: **R.id, R.layout, R.menu,...**

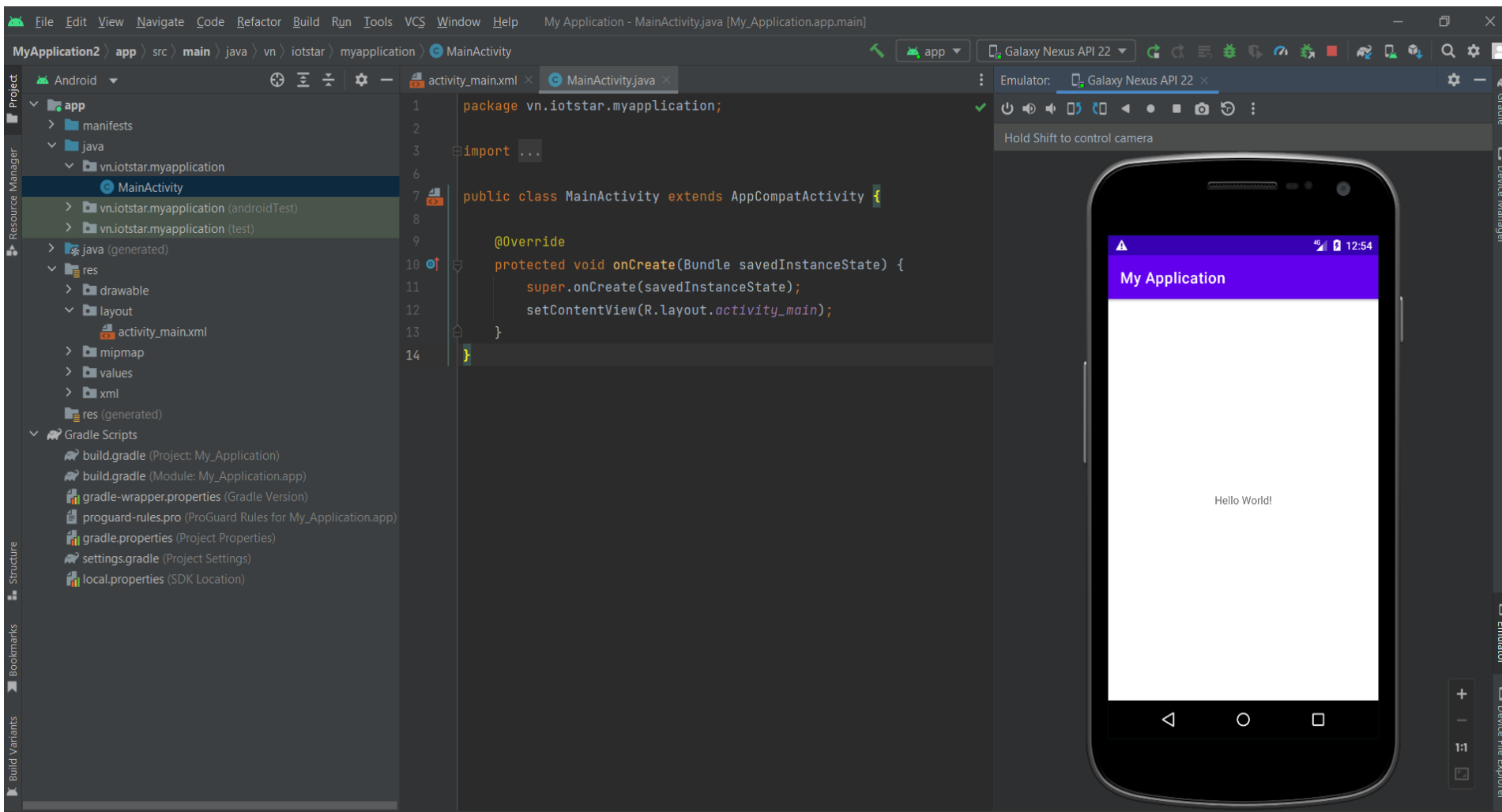
```
setContentView(R.layout.activity_main);
```

```
Toolbar toolbar = findViewById(R.id.toolbar);
```



- **Build.gradle:** thông tin cấu hình project, các dependency của project.





The screenshot shows the Android Studio IDE with the following components:

- Top Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help. The current file is MainActivity.java.
- Left Sidebar (Project Structure):**
  - Project: MyApplication2
  - app (selected)
    - manifests
    - java
      - vn.iotstar.myapplication
        - MainActivity (selected)
      - vn.iotstar.myapplication (androidTest)
      - vn.iotstar.myapplication (test)
    - java (generated)
    - res
      - drawable
      - layout
        - activity\_main.xml
      - mipmap
      - values
      - xml
    - res (generated)
  - Gradle Scripts
    - build.gradle (Project: My\_Application)
    - build.gradle (Module: My\_Application.app)
    - gradle-wrapper.properties (Gradle Version)
    - proguard-rules.pro (ProGuard Rules for My\_Application.app)
    - gradle.properties (Project Properties)
    - settings.gradle (Project Settings)
    - local.properties (SDK Location)

- Center Editor:** MainActivity.java

```

1 package vn.iotstar.myapplication;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }

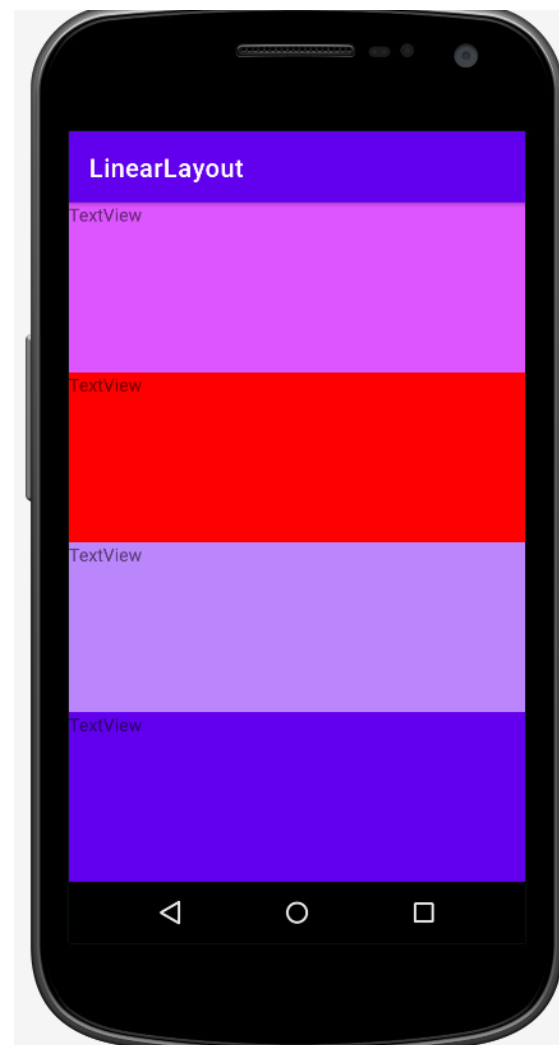
```
- Right Sidebar:**
- Emulator: Galaxy Nexus API 22
- Hold Shift to control camera
- Device Manager
- Emulator
- Device File Explorer

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);//will hide the title not the title bar

    this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);//int flag, int mask

    setContentView(R.layout.activity_main);
}
```





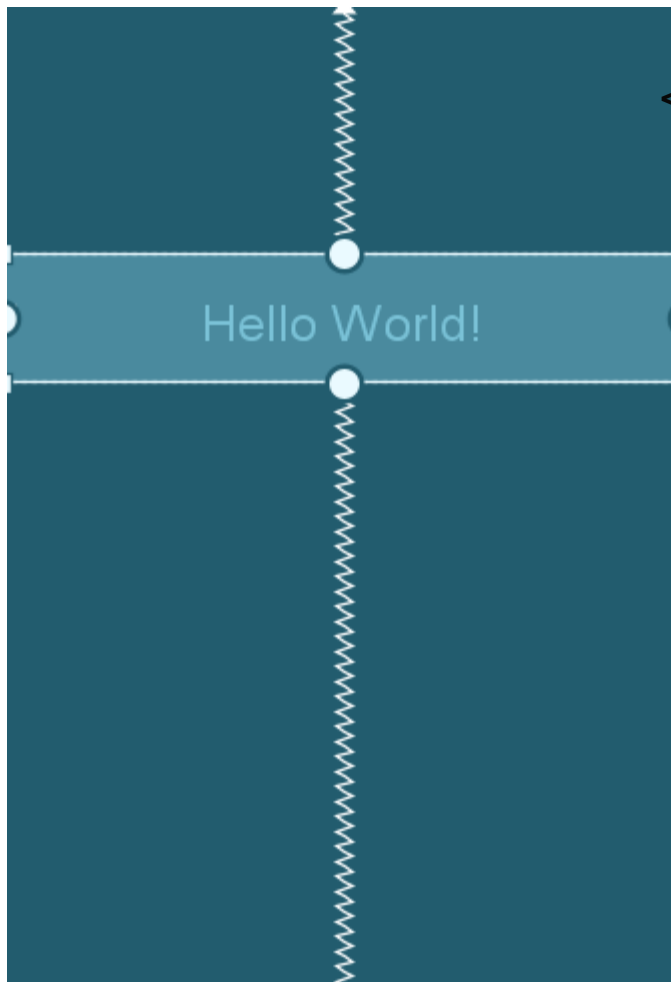
## QUY TRÌNH THIẾT KẾ ỨNG DỤNG

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



- Quy trình thiết kế ứng dụng
- Ví dụ minh họa

- Bước 1 Kéo thả: thiết kế giao diện (Layout, Control)
- Bước 2 Ảnh xạ: các view trên giao diện
- Bước 3 Viết code: điều khiển các view và bài toán.



<TextView

```

    android:id="@+id/textView1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:text="Hello World!"
    android:textAlignment="center"
    android:textColor="#FF9800"
    android:textSize="34sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.44"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.234" />
    
```

```
public class MainActivity extends AppCompatActivity {
```

```
    TextView txtNoiDung1; //khai báo toàn cục
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        //ánh xạ dựa vào Id
```

```
        txtNoiDung1 = (TextView) findViewById(R.id.textView1);
```

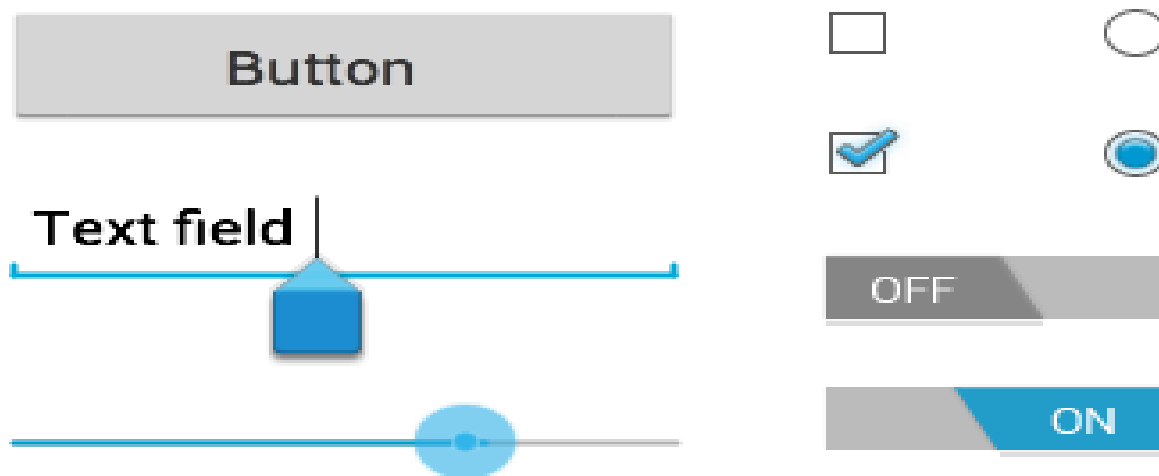
```
        //viết code điều khiển textView1
```

```
        txtNoiDung1.setText("Chào bạn");
```

```
    }
```

```
}
```

- Input Control là các thành phần có tính tương tác trong giao diện UI của ứng dụng. Android cung cấp nhiều control đa dạng để bạn có thể sử dụng trong UI như button, text field, seek bar, checkbox, zoom button, toggle button, ...



- **TextView** được sử dụng để hiển thị text tới người dùng.

```
<TextView android:id="@+id/text_id"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World" />
```

```
TextView myText = (TextView) findViewById(R.id.text_id);
```

- EditText Là một lớp con được định nghĩa trước của TextView mà bao gồm các khả năng chỉnh sửa đa dạng.

```
<EditText
    android:id="@+id/textview"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:inputType="text"
    android:maxLines="1"
    android:lines="1"/>
```

Thuộc tính **android:inputType** dùng để thiết lập kiểu nhập liệu, hiện thị

- **AutoCompleteTextView** trong Android hỗ trợ cho người dùng những gợi ý liên quan khi bạn nhập vào trường EditText.

```

java.lang.Object
└─ android.view.View
    └─ android.widget.TextView
        └─ android.widget.EditText
            └─ android.widget.AutoCompleteTextView
    
```

```

<AutoCompleteTextView
android:id="@+id/autoCompleteTextView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/textView1"
android:layout_marginLeft="36dp"
android:layout_marginTop="17dp"
android:ems="10"
android:text="">
<requestFocus />
</AutoCompleteTextView>
    
```



```
<Button
    android:id="@+id/btnThucHien"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="180dp"
    android:text="Thực hiện"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1"
    app:layout_constraintVertical_bias="0.353" />
```

```
//viết code điều khiển btnClick
btnClick.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //viết code điều khiển textView1
        txtNoiDung1.setText("Chào bạn");
    }
});
```

UserName

Password

**ĐĂNG NHẬP**

<EditText

```

    android:id="@+id/editTextPassword"
    android:layout_width="345dp"
    android:layout_height="51dp"
    android:layout_marginTop="12dp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextName" />
    
```

<EditText

```

    android:id="@+id/editTextName"
    android:layout_width="345dp"
    android:layout_height="51dp"
    android:layout_marginTop="32dp"
    android:ems="10"
    android:hint="UserName"
    android:inputType="textPersonName"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"/>
    
```

UserName

Password

**DĂNG NHẬP**

<Button

```

    android:id="@+id/btnDangNhap"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:text="Đăng nhập"
    android:textColor="#fff"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.86"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf=
"@+id/editTextPassword" />

```

```
btnDangnhap =(Button) findViewById(R.id.btnDangNhap);
editTextname = (EditText) findViewById(R.id.editTextName);
editTextpass = (EditText) findViewById(R.id.editTextPassword);
```

```
btnDangnhap.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //viết code
        if(editTextname.getText().toString().equals("admin")
            && editTextpass.getText().toString().equals("admin")){
            txtNoiDung1.setText("Đăng nhập thành công!");
        }else{
            txtNoiDung1.setText("Đăng nhập thất bại" +
editTextpass.getText().toString());
        }
    }
});
```



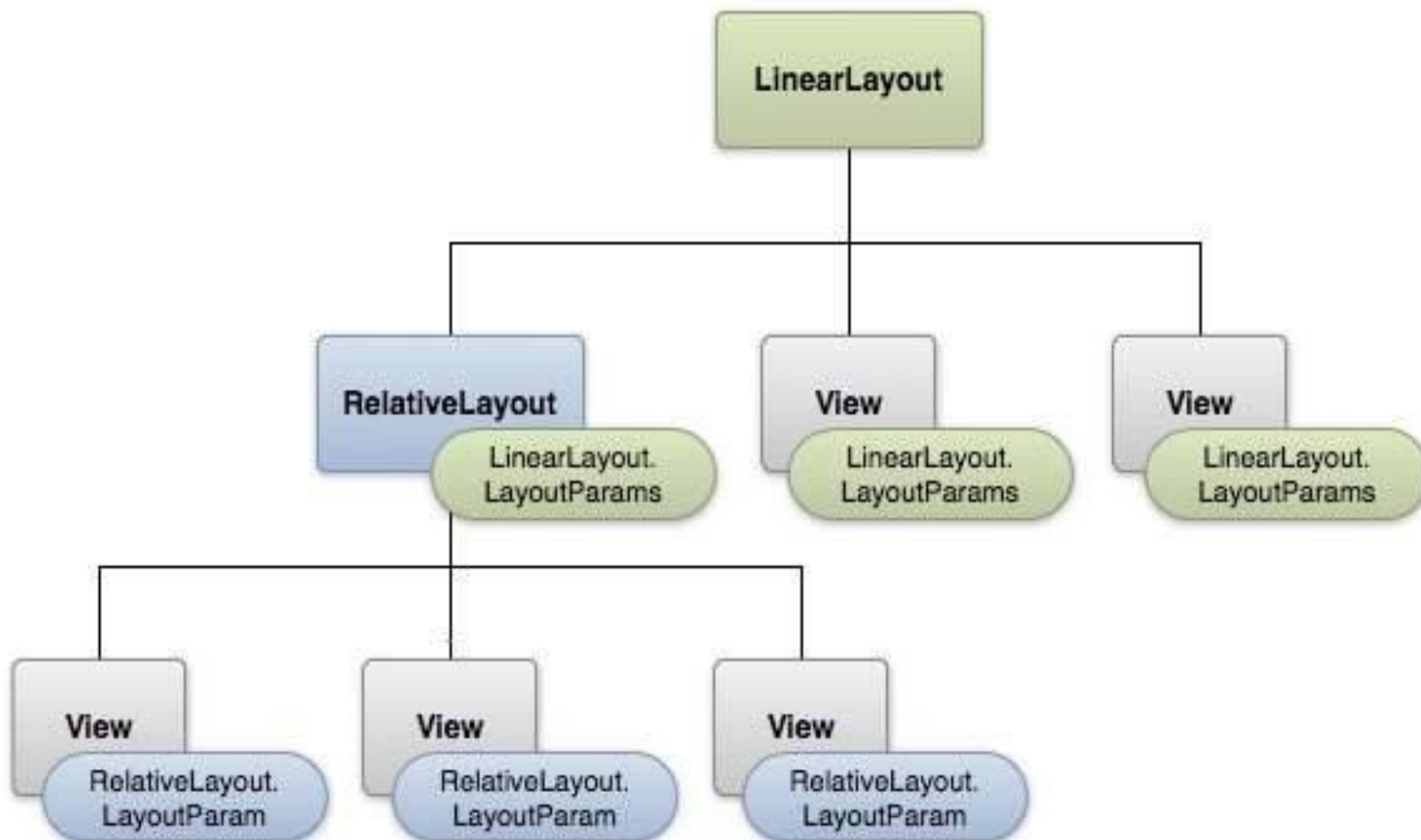
## LAYOUT TRONG ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



- Kiến trúc nền tảng cho giao diện UI là một đối tượng View, view là lớp cơ sở cho Widget để tạo các thành phần UI có tính tương tác như button, các trường text, ...
- **ViewGroup** là một lớp cung cấp Container giữ các View khác hoặc các ViewGroup khác và định nghĩa các thuộc tính Layout của chúng.

- Layout đặc trưng định nghĩa cấu trúc nhìn thấy cho một giao diện UI trong Android và có thể được tạo bởi sử dụng các đối tượng **View/ViewGroup** hoặc bạn có thể khai báo Layout của bạn bởi sử dụng XML file đơn giản là **main\_Layout.xml**, được đặt trong thư mục **res/layout** của Project.





Layout	Miêu tả
<b>Linear Layout</b>	Linear Layout là một view group mà căn chỉnh các view con theo một hướng nào đó: <b>chiều dọc hay chiều ngang</b> . Thuộc tính: <b>android:orientation</b>
<b>Relative Layout</b>	Là Layout hiển thị các View con với các vị trí tương đối. Vị trí của mỗi View có thể được xác định so với các View khác hoặc với thành phần cha của chúng (thông qua id). Bạn có thể sắp xếp View sang bên phải, bên dưới một View khác, giữa màn hình, v.v.. Để định nghĩa vị trí cho mỗi View bạn sử dụng nhiều thuộc tính có sẵn từ <b>RelativeLayout.LayoutParams</b> .
<b>Table Layout</b>	Table Layout là một view mà nhóm tất cả các view vào <b>trong các hàng và các cột</b> .

Layout	Công dụng
<b>Absolute Layout</b>	Là layout cho phép bạn chỉ định vị trí chính xác ( tọa độ x / y) của các đối tượng. Absolute layouts thường ít linh hoạt và khó duy trì hơn các loại layouts khác.
<b>Frame Layout</b>	FrameLayout là một trong những layout hữu ích được cung cấp bởi hệ thống Android, cho phép các đối tượng giao diện người dùng được xếp chồng chéo với nhau.

Layout	Công dụng
<b>Constraint layout</b>	<p>Constraint layout được giới thiệu lần đầu tiên tại sự kiện <b>Google I/O 2016</b>, Constraint Layout sẵn dùng với bản Android 2.3 (API level 9)</p> <p><b>match_constraint</b>: xác định width và height của một View</p> <p>VD : TextView dưới được set nằm giữa view parent</p> <pre>&lt;TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Text View 1" app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintLeft_toLeftOf="parent" app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" /&gt;</pre>

Thuộc tính	Ý nghĩa
<b>android:id</b> <b>android:id="@+id/mybutton"</b>	ID nhận diện duy nhất View, Button myButton = (Button) findViewById(R.id.mybutton);
<b>android:layout_width</b>	Độ rộng của Layout, dp (Density-independent pixel), sp (Scale-independent pixel), pt (các point là 1/72 inch), px (pixel), mm (mili met), và in (inch).
<b>android:layout_height</b>	Chiều cao của Layout, <b>wrap_content</b> theo nội dung, <b>fill_parent/match_parent</b> theo view cha.
<b>android:layout_marginTop</b>	Lề cạnh trên của Layout
<b>android:layout_marginBottom</b>	Lề cạnh dưới của Layout
<b>android:layout_marginLeft</b>	Lề cạnh trái của Layout
<b>android:layout_marginRight</b>	Lề cạnh phải Layout
<b>android:layout_gravity</b>	Xác định cách các view con được đặt tại đâu

Thuộc tính	Ý nghĩa
<b>android:layout_weight</b>	Xác định có bao nhiêu phần trong Layout được cấp phát tới View đó
<b>android:layout_x</b>	Xác định tọa độ x của Layout
<b>android:layout_y</b>	Xác định tọa độ y của Layout
<b>android:paddingLeft</b>	Left padding được điền cho Layout
<b>android:paddingRight</b>	Right padding được điền cho Layout
<b>android:paddingTop</b>	Top padding được điền cho Layout
<b>android:paddingBottom</b>	Bottom padding được điền cho Layout

```

android:layout_marginLeft="40dp"
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="40dp"
    android:layout_marginTop="30dp"
    android:background="#72acdf"
    android:paddingBottom="50dp"
    android:paddingLeft="50dp"
    android:paddingRight="50dp"
    android:paddingTop="50dp"
    tools:context="dev4u.com.helloworld.MainActivity">

    <Button
        android:id="@+id/btnMyButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button Demo" />

    <!--<TextView-->
    <!--android:layout_width="wrap_content"-->
    <!--android:layout_height="wrap_content"-->
    
```





The screenshot displays the Android Studio IDE. On the left, the 'Project' view shows the file structure of 'LayoutDemo'. On the right, the 'Manifest' view shows the contents of 'AndroidManifest.xml'.

**Project Structure (Left):**

- LayoutDemo
  - src
    - com.example.layoutdemo
      - MainActivity.java (1)
  - gen [Generated Java Files]
  - Android 7.1.1
  - Android Private Libraries
  - Android Dependencies
  - assets
  - bin
  - libs
  - res
    - drawable-hdpi
    - drawable-ldpi
    - drawable-mdpi
    - drawable-xhdpi
    - drawable-xxhdpi
    - layout
      - activity\_main.xml (2)
    - menu
    - values
      - values-v11
      - values-v14
      - values-w820dp
    - AndroidManifest.xml (3)
    - ic\_launcher-web.png
    - proguard-project.txt
    - project.properties

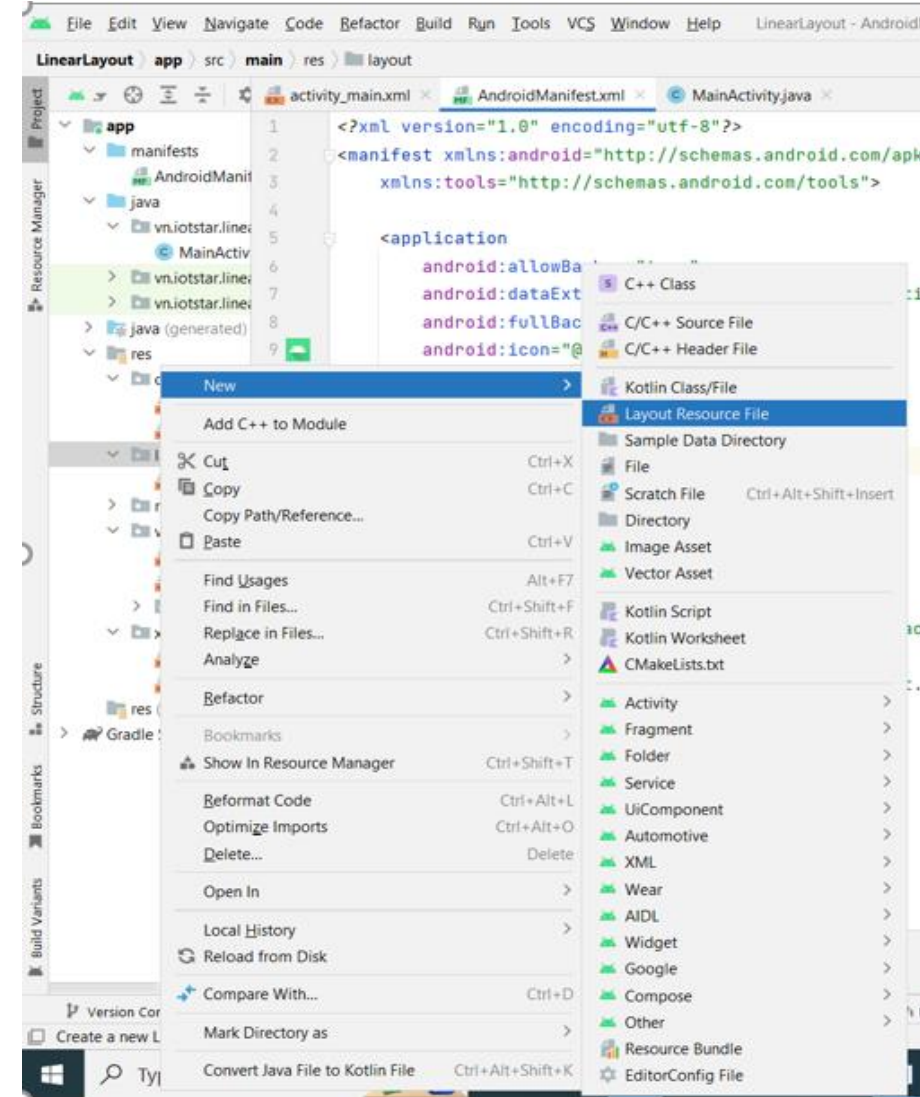
**AndroidManifest.xml (Right):**

```

4  android:versionCode="1"
5  android:versionName="1.0" >
6
7  <uses-sdk
8      android:minSdkVersion="8"
9      android:targetSdkVersion="21" />
10
11 <application
12     android:allowBackup="true"
13     android:icon="@drawable/ic_launcher"
14     android:label="@string/app_name"
15     android:theme="@style/AppTheme" >
16     <activity
17         android:name=".MainActivity"
18         android:label="@string/app_name" >
19         <intent-filter>
20             <action android:name="android.intent.action.MAIN" />
21
22             <category android:name="android.intent.category.LAUNCHER" />
23         </intent-filter>
24     </activity>
25 </application>
26
27 </manifest>
28
  
```

Callout 4 points to the activity declaration block (lines 16-24).





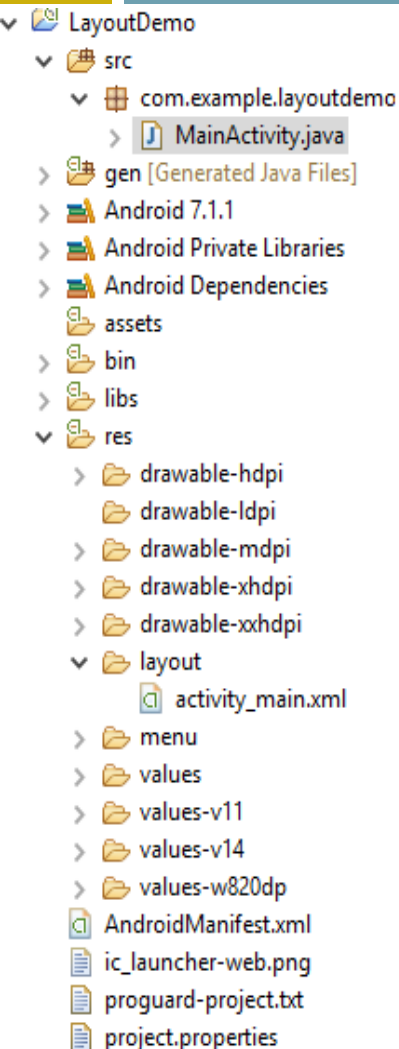
Đặt tên my\_new\_layout.xml

Bây giờ vào lại MainActivity.java. Sửa lại dòng lệnh setContentView thành:

```
protected void onCreate(Bundle
    savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.my_new_layout);
}
```



65



```

5 import android.view.Menu;
6 import android.view.MenuItem;
7
8 public class MainActivity extends Activity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15
16     @Override
17     public boolean onCreateOptionsMenu(Menu menu) {
18         // Inflate the menu; this adds items to the action bar if it is present.
19         getMenuInflater().inflate(R.menu.main, menu);
20         return true;
21     }
22
23     @Override
24     public boolean onOptionsItemSelected(MenuItem item) {
25         // Handle action bar item clicks here. The action bar will
26         // automatically handle clicks on the Home/Up button, so long
27         // as you specify a parent activity in AndroidManifest.xml.
28         int id = item.getItemId();
29         if (id == R.id.action_settings) {
30             return true;
31         }
32         return super.onOptionsItemSelected(item);
33     }
34 }

```

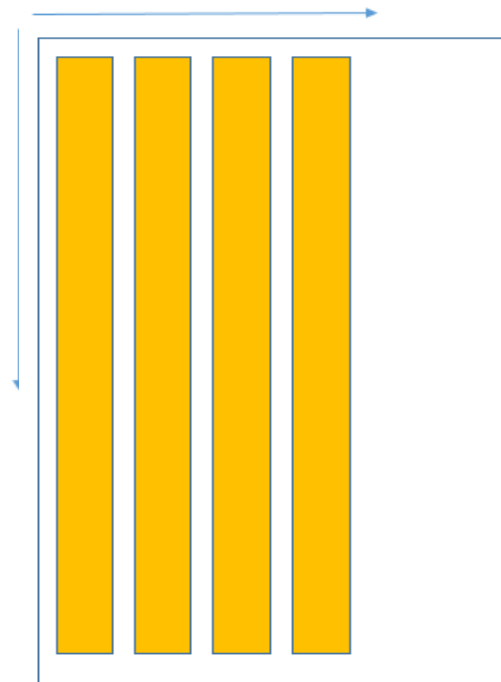
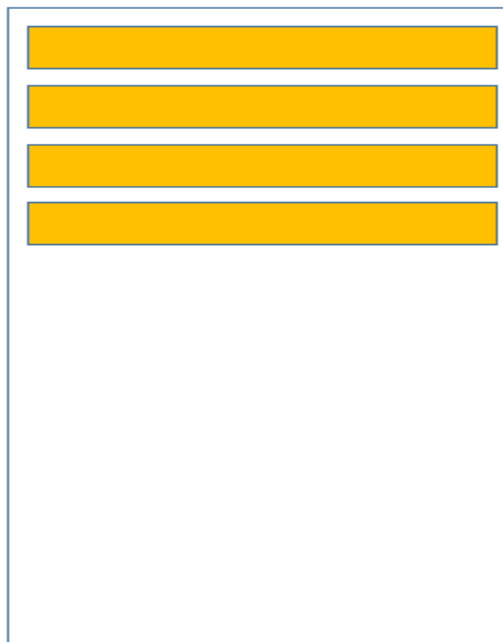
FrameLayout là một ViewGroup được sử dụng rất nhiều trong android. Bởi vì nó là ViewGroup đơn giản nhất,

FrameLayout được định nghĩa bắt đầu bởi thẻ <FrameLayout> và thẻ đóng </FrameLayout>. Ở giữa thẻ đóng và thẻ mở chính là các view con của nó.

Quy tắc layout các view con trong FrameLayout là các view sẽ nằm chồng lên nhau, view thêm vào sau sẽ nằm đè lên view nằm phía dưới

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/FrameLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textColor="#2c3e50"
        android:textSize="32sp" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher_background"/>
</FrameLayout>
```

- LinearLayout là một view group việc căn chỉnh các view con theo một hướng nào đó: chiều dọc hay chiều ngang.
- Khi Layout được tạo, bạn có thể tải layout resource từ phần code, trong phương thức callback là **Activity.onCreate()**, như sau:



```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

```
android:orientation="vertical"
android:orientation="horizontal"
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<TextView android:id="@+id/text"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a TextView" />
```

```
<Button android:id="@+id/button"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is a Button" />
```

```
<!-- More GUI components go here -->
```

```
</LinearLayout>
```

<TextView

```
android:background="#ff0"
android:fontFamily="sans-serif"
android:textColor="#FF5722"
android:textSize="30sp"
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World"/>
```

<TextView

```
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:background="#af5656" />
```

<TextView

```
android:layout_width="0dp"
android:layout_height="match_parent"
android:layout_weight="4"
android:background="#28c6a6" />
```

## Chia tỉ lệ layout

Ngoài cách sử dụng **LinearLayout** thông thường thì **LinearLayout** thường được sử dụng để phân chia tỉ lệ layout bằng cách sử dụng thuộc tính **layout\_weight** và **weightSum**.

- **weightSum**: Xác định trọng số của **LinearLayout** hiện tại.
- **layout\_weight**: Trọng số mà view con trong **LinearLayout** chiếm giữ.

<TextView

android:background="#ff0"

android:fontFamily="sans-serif"

android:textColor="#FF5722"

android:textSize="30sp"

android:id="@+id/textView5"

android:layout\_width="match\_parent"

android:layout\_height="100dp"

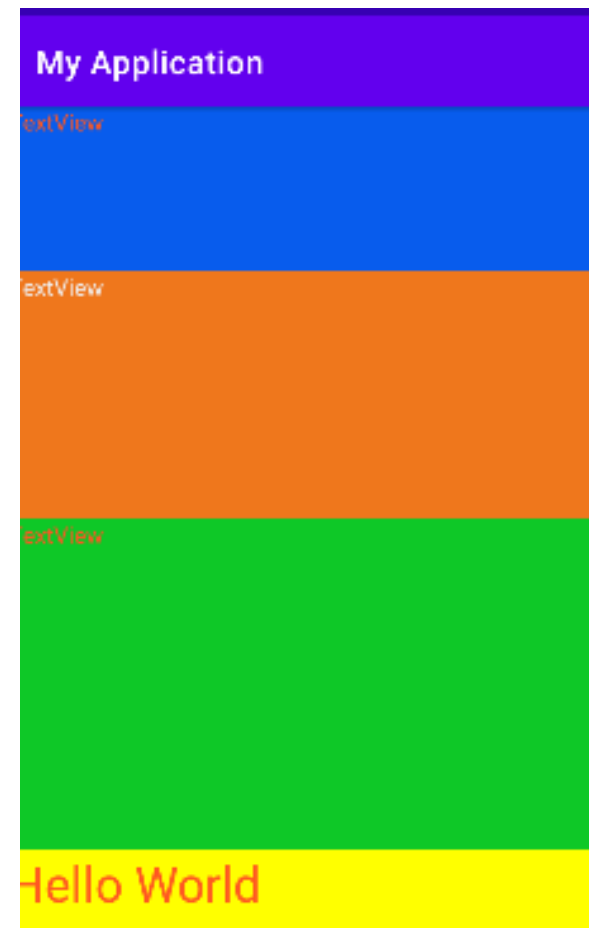
android:text="Hello World"

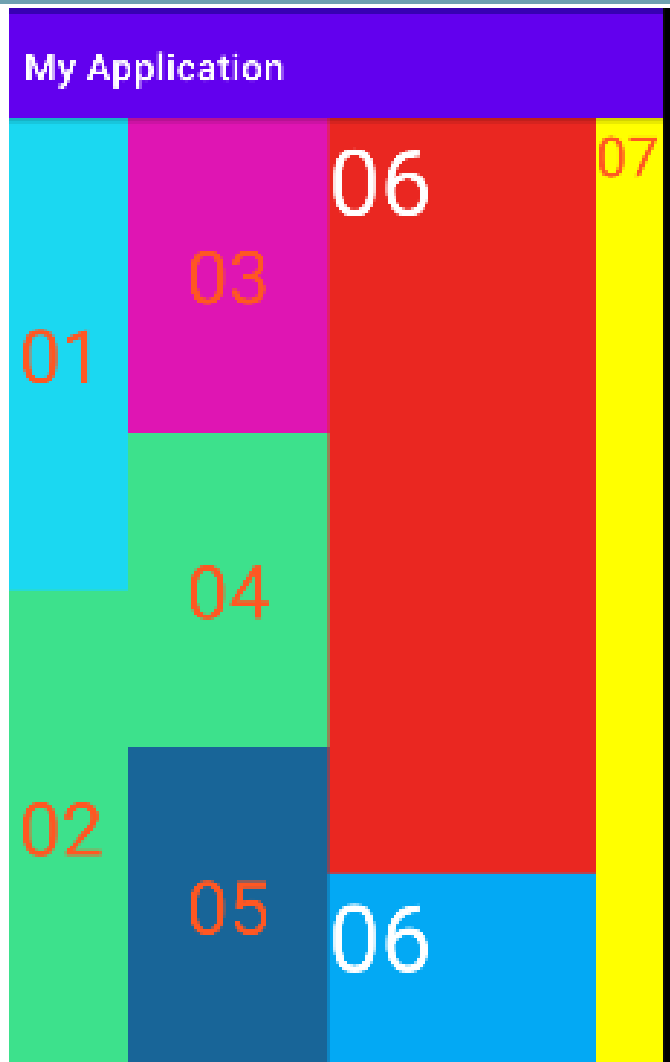
android:layout\_weight="1" //chia trọng số, tỉ lệ

/>

<LinearLayout

android:weightSum="10" /> <!-- 2 - 3 - 4 -1 -->





- RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối. Do đó khi làm RelativeLayout bạn phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn (do đó nếu đổi ID thì phải đổi luôn các tham chiếu khác sao cho khớp với Id bạn mới đổi).
- RelativeLayout là ViewGroup cũng được sử dụng khá nhiều trong android, RelativeLayout được định nghĩa trong xml bởi cặp thẻ đóng mở **<RelativeLayout>** và thẻ đóng **</RelativeLayout>**.



Quy tắc layout của RelativeLayout khá giống như FrameLayout, nhưng có một số điểm đặc biệt là các view có thể xác định bằng vị trí **tương đối (relative)** với các view khác thông qua id.

Các vị trí tương đối này như sau:

- **android:layout\_above="id\_name"**: view hiện tại sẽ nằm phía trên view có thuộc tính id là id\_name.
- **android:layout\_below="id\_name"**: view hiện tại sẽ nằm phía dưới view có thuộc tính id là id\_name.
- **android:layout\_toLeftOf="id\_name"**: view hiện tại sẽ nằm bên trái dưới view có thuộc tính id là id\_name.
- **android:layout\_toRightOf="id\_name"**: view hiện tại sẽ nằm phía bên phải view có thuộc tính id là id\_name.

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
3. android:layout_width="match_parent"
4. android:layout_height="match_parent">
5.
6. <TextView
7. android:id="@+id/tv_username"
8. android:layout_width="wrap_content"
9. android:layout_height="wrap_content"
10.    android:text="username" />
11.
12. <EditText
13. android:id="@+id/edt_username"|
14. android:layout_width="match_parent"
15. android:layout_height="wrap_content"
16. android:layout_below="@+id/tv_username" />
17.
18. <TextView
19. android:id="@+id/tv_password"
20. android:layout_width="wrap_content"
21. android:layout_height="wrap_content"
22. android:layout_below="@id/edt_username"
23. android:text="password" />
24.
25. <EditText
26. android:id="@+id/edt_password"
27. android:layout_width="match_parent"
28. android:layout_height="wrap_content"
29. android:layout_below="@id/tv_password"
30. android:inputType="textPassword" />
31.
32. </RelativeLayout>

```

Một số thuộc tính khác thường xuyên sử dụng với RelativeLayout:

- **android:layout\_alignParentBottom="boolean"**  
: Căn dưới phần tử hiện theo phần tử cha nếu set là true.
- **android:layout\_alignParentTop="boolean"**: Căn trên phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout\_alignParentRight="boolean"**: Căn phải phần tử hiện tại theo phần tử cha nếu set là true.
- **android:layout\_alignParentLeft="boolean"**: Căn trái phần tử hiện tại theo phần tử cha nếu set là true.

- Table Layout là một view mà nhóm tất cả các view vào **trong các hàng và các cột.**

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<TableLayout
```

```
    android:stretchColumns="*"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <TableRow>
```

```
        <ImageView
```

```
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:src="@drawable/android"/>
```

```
        <TextView
```

```
            android:layout_span="3"
            android:layout_gravity="center"
            android:textSize="30sp"
            android:text="dong 1"/>
```

```
        </TextView>
```

```
    </TableRow>
```

```
</TableLayout>
```

**Các thuộc tính quan trọng:**

- android:stretchColumns
- android:shrinkColumns
- android:collapseColumns

Dùng **layout\_span** để trộn các cột

```
<TableRow>
    <TextView android:text="URL:" />
    <EditText
        android:id="@+id/entry"
        android:layout_span="3" />
</TableRow>
```

- ❑ Constraint layout được giới thiệu lần đầu tiên tại sự kiện **Google I/O 2016**, Constraint Layout sẵn dùng với bản Android 2.3 (API level 9)

❑ **match\_constraint**: xác định width và height của một View

- ❑ VD : TextView dưới được set nằm giữa view parent

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Text View 1"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintLeft_toLeftOf="parent"
```

```
    app:layout_constraintRight_toRightOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />
```

- Thực hiện 02 điểm neo: ở trên bên trái và ở trên bên phải (hoặc ở dưới bên trái và ở dưới bên phải)

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.206" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="88dp"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.233"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf
    ="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout
>
```



## Điều khiển sự kiện (Event - Handling)

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



Có 3 khái niệm liên quan đến quản lý sự kiện:

- **Event Listeners:** Là một giao diện trong một View, View này chứa một phương thức duy nhất là callback. Phương thức này được gọi bởi Framework Android, khi View đã được đăng ký sự kiện và người dùng tác động lên View trong UI (giao diện người dùng).
- **Event Listeners Registration:** Event Registration là một quá trình mà một Event Handler đã được đăng ký với Event Listener để mà Handler này được gọi khi Event Listener kích hoạt sự kiện.
- **Event Handlers:** Khi một Event xảy ra, và chúng ta đã đăng ký một Event Listener cho sự kiện, thì Event Listener gọi Event Handler, là phương thức mà thực sự xử lý sự kiện đó

Event Handler	Event Listener Name
<b>onClick()</b>	<b>OnClickListener()</b> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focuse trên bất kỳ widget như button, text, image vv. Chúng ta sẽ sử dụng phương onClick() để xử lý sự kiện.
<b>onLongClick()</b>	<b>OnLongClickListener()</b> Đăng ký sự kiện khi người dùng hoặc click hoặc chạm (touche) hoặc focus trên bất kỳ widget như button, text, image vv. trong một hoặc nhiều giây. Chúng ta sẽ sử dụng phương onLongClick() để xử lý sự kiện.
<b>onFocusChange()</b>	<b>OnFocusChangeListener()</b> Sự kiện phát sinh khi widget mất focus.



Event Handler	Event Listener Name
<b>onKey()</b>	OnFocusChangeListener() Sự kiện phát sinh khi người dùng focus trên widget và nhấn (presse) hoặc thả (release) một phím trên thiết bị.
<b>onTouch()</b>	OnTouchListener() Sự kiện phát sinh khi người dùng nhấn phím, nhả phím, hoặc bắt kỳ cử chỉ chuyển động trên màn hình.
<b>onMenuItemClick()</b>	OnMenuItemClickListener() Sự kiện phát sinh khi người dùng chọn một mục trong menu.
<b>onCreateContextMenu()</b>	OnCreateContextMenuListener() Sự kiện phát sinh khi người dùng chọn một mục trong menu ngữ cảnh (Context Menu)

1. Xử lý sự kiện trong Layout (Handle event in Layout)
2. Xử lý sự kiện bằng lớp nặc danh (Inline anonymous listener)
3. Kế thừa Interface OnClickListener (Implements OnClickListener Interface)
4. Sử dụng biến (Event Listener using Variable)
5. Bắt sự kiện thông qua lớp lắng nghe (Listener Class)
6. PerformClick method

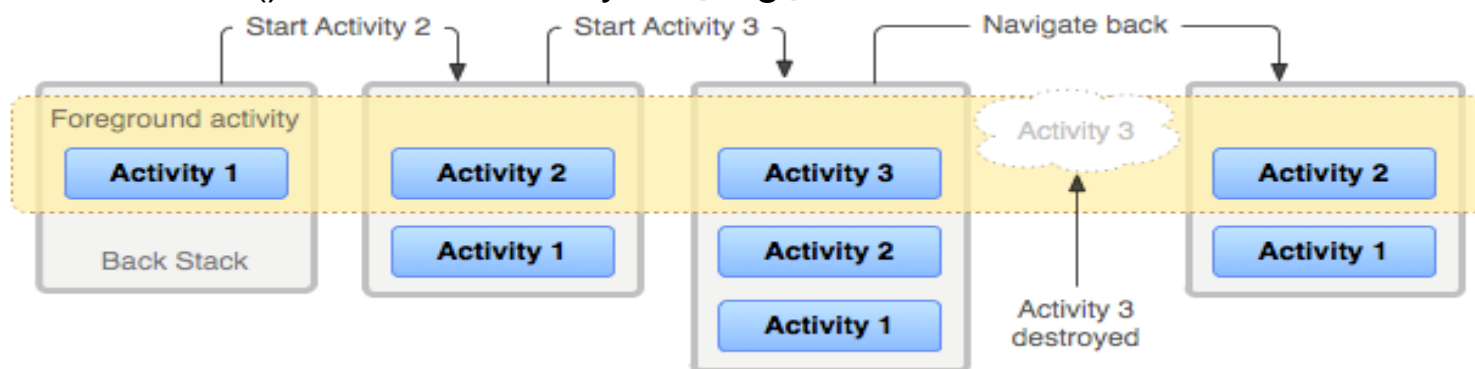


## CHUYỂN ACTIVITY TRONG ANDROID

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

## □ Không gửi kèm dữ liệu:

- Khi chuyển sang KetQuaActivity thì MainActivity được đẩy xuống dưới Back Stack và KetQuaActivity sẽ ở trên đầu của Back Stack. Vì vậy khi nhấn trở lại từ KetQuaActivity thì MainActivity được đẩy lên và onResume() của MainActivity được gọi.



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

Mọi Activity tạo ra phải được khai báo trong file **AndroidManifest.xml**, nếu không khai báo mà vẫn sử dụng Activity này thì hệ thống sẽ ném ra ngoại lệ, điều này cũng được áp dụng khi sử dụng Service trong Android.

```
<activity android:name=".KetQuaActivity"></activity>
```

## □ Gửi kèm dữ liệu

- ▣ Khi tạo Intent phải đặt các dữ liệu kèm theo qua đối tượng intent này, sử dụng các phương thức có tiền tố là put của đối tượng intent để gửi dữ liệu, mỗi dữ liệu đặt vào sẽ có 1 key đi kèm.
- ▣ Char, boolean, int, float, long, String hay kiểu mảng boolean[], char[], int[] , kiểu dữ liệu object, tuy nhiên những kiểu dữ liệu này phải implement interface Parcelable hoặc interface Serializable.

```
Intent intent = new Intent(MainActivity.this, KetQuaActivity.class);
intent.putExtra("int_key", 4);
intent.putExtra("char_key", 'r');
intent.putExtra("boolean_key", true);
intent.putExtra("long_key", 323L);
intent.putExtra("float_key", 3.2f);
intent.putExtra("string_key", "Chuyen Activity trong Android");
intent.putExtra("double", 32D);
intent.putExtra("int_array_key", new int[]{1, 2, 9});
intent.putExtra("boolean_array_key", new boolean[]{true, false, true, true});
intent.putExtra("char_array_Key", new char[] {'e', 'i', 't', 'g', 'u', 'i', 'd', 'e'});
intent.putExtra("rect_key", new Rect(0,0, 200, 200));
```

## □ Gửi dữ liệu đi sử dụng Bundle

- ▣ Thay vì sử dụng bằng cách đặt dữ liệu vào intent thì sử dụng 1 đối tượng gọi là **Bundle**, đặt các dữ liệu vào đối tượng này và sử dụng phương thức **putExtras()** để đặt hết vào intent.

```
Intent intent = new Intent(MainActivity.this, KetQuaActivity.class);
Bundle bundle = new Bundle();
bundle.putChar("char", 'e');
bundle.putInt("int", 3);
bundle.putString("string", "Nguyễn Hữu Trung");
bundle.putFloat("float", 5.2f);
bundle.putDouble("double", 843D);
bundle.putLong("long", 55343L);
bundle.putParcelable("parcelable", new Rect(0, 0, 300, 300));
intent.putExtras(bundle);
startActivity(intent);
```

## □ Nhận dữ liệu

- Sau khi startActivity(), ở Activity B (KetQuaActivity) có được dữ liệu mà Activity A (MainActivity) gửi đến như sau:
  - Lấy ra đối tượng Intent mà Activity A gửi qua bằng phương thức getIntent().
  - Từ đối tượng intent có được từ getIntent(), lấy những dữ liệu trong đối tượng này bằng phương thức getXExtra() (với X là kiểu dữ liệu muốn lấy).

```
// Get data from MainActivity
int intValue = intent.getIntExtra("int_key", 0);
char charValue = intent.getStringExtra("char_key", 'a');
float floatValue = intent.getFloatExtra("float_key", 0f);
boolean booleanValue = intent.getBooleanExtra("boolean_key", false);
String stringValue = intent.getStringExtra("string_key");
int[] intArrayValue = intent.getIntArrayExtra("int_array_key");
char[] charArrayValue = intent.getCharArrayExtra("char_array_key");
Rect rect = intent.getParcelableExtra("rect_key");
```

## □ Nhận dữ liệu

- Sau khi startActivity(), ở Activity B (KetQuaActivity) có được dữ liệu mà Activity A (MainActivity) gửi đến như sau:
  - Lấy ra đối tượng Intent mà Activity A gửi qua bằng phương thức getIntent().
  - Từ đối tượng intent có được từ getIntent(), lấy những dữ liệu trong đối tượng này bằng phương thức getXExtra() (với X là kiểu dữ liệu muốn lấy).

Nếu gửi dữ liệu bằng **Bundle** thì làm như sau:

```
Intent intent = getIntent();
Bundle bundle = intent.getExtras();
char charValue = bundle.getChar("char");
int intValue = bundle.getInt("int");
String stringValue = bundle.getString("string");
float floatValue = bundle.getFloat("float");
double doubleValue = bundle.getDouble("double");
long longValue = bundle.getLong("long");
Rect rect = bundle.getParcelable("parcelable");
```



# KHOA CÔNG NGHỆ THÔNG TIN



## LISTVIEW

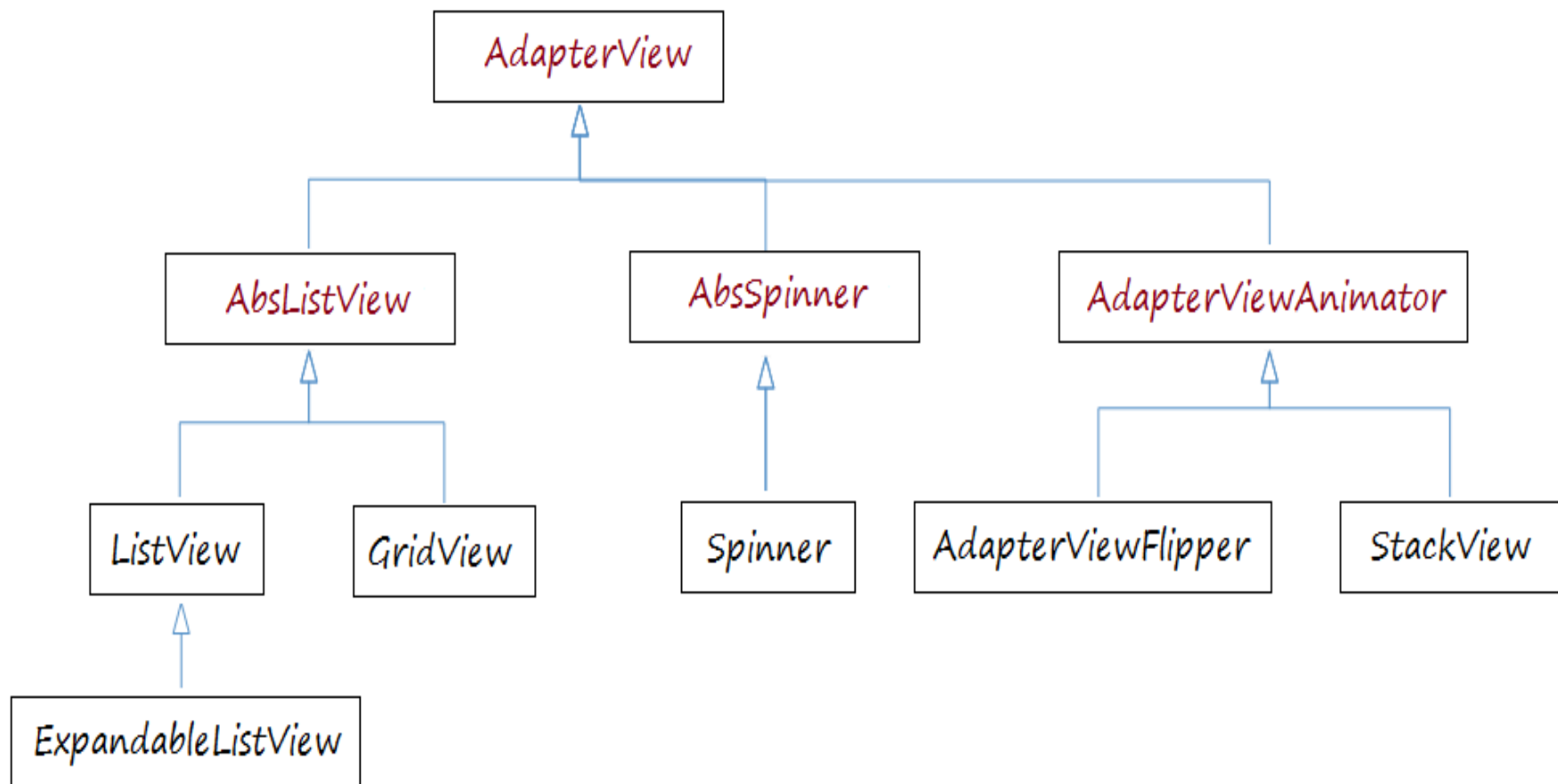
Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM

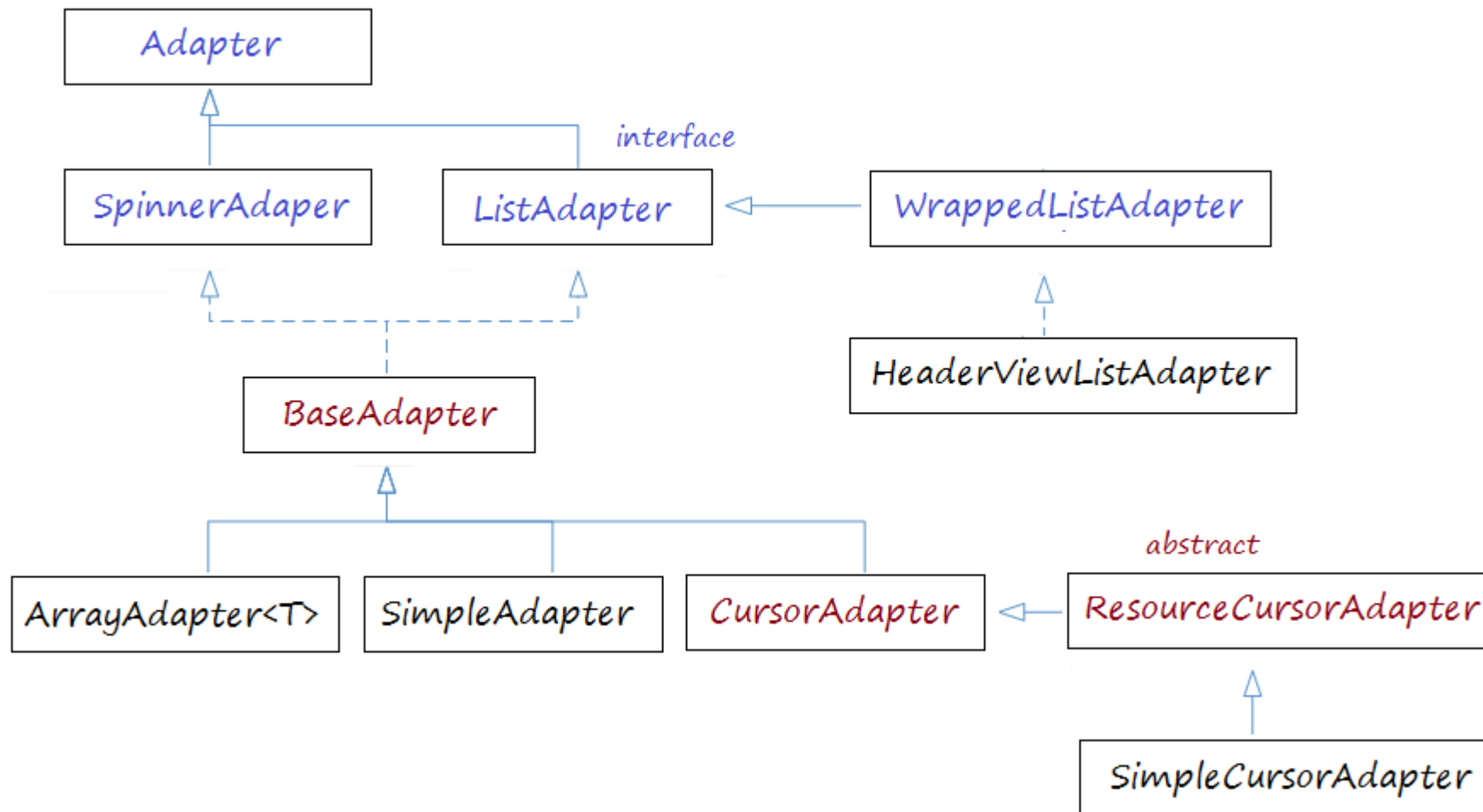


ThS. Nguyễn Hữu Trung



- Có nhiều View cần tới Android Adapter để quản lý dữ liệu hiển thị, các View này là con của class AdapterView

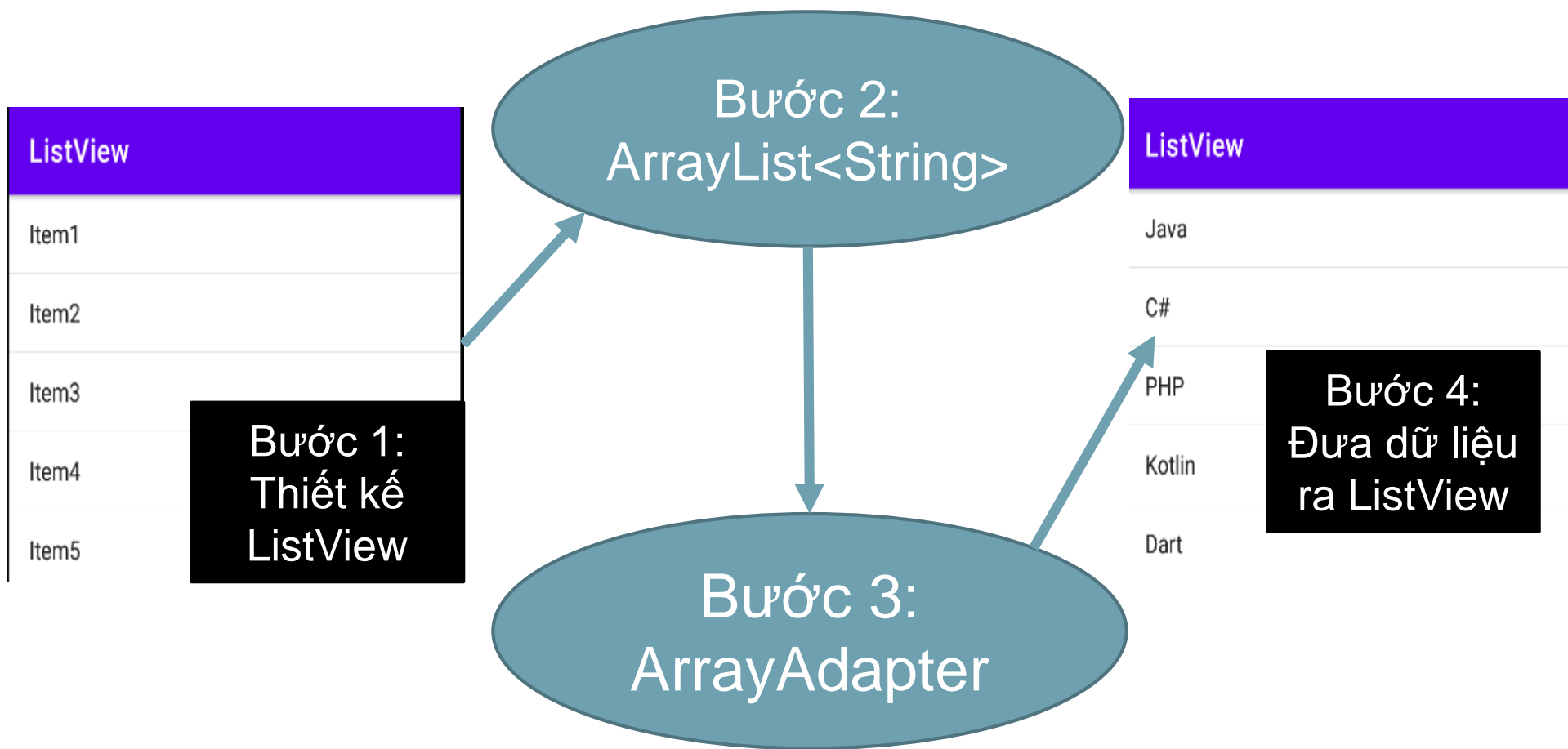




- **ListView** trong Android là một view group, hiển thị các item theo một danh sách có thể cuộn được theo chiều thẳng đứng.

```
<ListView
    android:id="@+id/listview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
</ListView>
```

Thuộc tính	Mô tả
<b>android:id</b>	ID là duy nhất để nhận diện Layout
<b>android:divider</b>	Nó có thể vẽ hoặc tô màu giữa các item trong danh sách.
<b>android:dividerHeight</b>	Nó xác định chiều cao của dải phân cách. Điều này có thể là px, dp, sp, in hoặc mm.
<b>android:entries</b>	Chỉ định tham chiếu đến một tài nguyên mảng sẽ điền vào ListView.
<b>android:footerDividersEnabled</b>	Khi được đặt thành false, ListView sẽ không vẽ dải phân cách trước mỗi chế độ xem chân trang. Giá trị mặc định là true.
<b>android:headerDividersEnabled</b>	Khi được đặt thành false, ListView sẽ không vẽ dải phân cách sau mỗi chế độ xem tiêu đề. Giá trị mặc định là true.



ListView

Item1

Item2

Item3

Item4

Item5

Bước 1:  
Thiết kế  
ListView

Mở file .xml thiết kế ListView

<ListView

android:id="@+id/listview1"

android:layout\_width="409dp"

android:layout\_height="354dp"

tools:layout\_editor\_absoluteX="1dp"

tools:layout\_editor\_absoluteY="1dp"

tools:ignore="MissingConstraints" />

## Bước 2: ArrayList<String>

```
public class MainActivity extends
AppCompatActivity {
    //khai báo
    ListView listView;
    ArrayList<String> arrayList;
```

*//ánh xạ*

```
listView = (ListView) findViewById(R.id.listView1);
```

*//Thêm dữ liệu vào List*

```
arrayList = new ArrayList<>();
```

```
arrayList.add("Java");
```

```
arrayList.add("C#");
```

```
arrayList.add("PHP");
```

```
arrayList.add("Kotlin");
```

```
arrayList.add("Dart");
```

## Bước 3: ArrayAdapter

- ArrayAdapter có 03 tham số:
- Context: màn hình hiển thị
  - Dạng Layout muốn đổ vào
  - Dữ liệu List đổ vào

*//Tạo ArrayAdapter*

```
ArrayAdapter adapter = new ArrayAdapter(
    MainActivity.this,
    android.R.layout.simple_list_item_1,
    arrayList
);
```



## ListView

Java

C#

PHP

Kotlin

Dart

**Bước 4:**  
Đưa dữ liệu  
ra ListView

*//truyền dữ liệu từ adapter ra listview*

**listView.setAdapter(adapter);**

## ListView

Java

C#

PHP

Kotlin

Dart

2

```
//bắt sự kiện click nhanh trên từng dòng của
//Listview
listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?>
adapterView, View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên ListView -> i
        //ban đầu =0
        Toast.makeText(MainActivity.this, ""+i,
        Toast.LENGTH_SHORT).show();
    }
});
```

Lấy nội dung: `arrayList.get(i)`

## ListView

Java

C#

PHP

Kotlin

Dart

2

```
//bắt sự kiện click giữ trên từng dòng của Listview
listView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean
onItemLongClick(AdapterView<?> adapterView,
View view, int i, long l) {
    //code yêu cầu
    //i: trả về vị trí click chuột trên ListView -> i
    ban đầu =0
    Toast.makeText(MainActivity.this,"Bạn đang
nhấn giữ "+ i + "-" + arrayList.get(i) ,
Toast.LENGTH_SHORT).show();
    return false;
}
});
```

ListView

Thêm môn học:

THÊM

ASP.NET

Java

C#

PHP

Kotlin

Dart

```
EditText editText1;  
Button btnNhap;
```

```
editText1 = (EditText) findViewById(R.id.editText1);  
btnNhap = (Button) findViewById(R.id.btnNhap);
```

```
btnNhap.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String name = editText1.getText().toString();  
        arrayList.add(name);  
        adapter.notifyDataSetChanged();  
    }  
});
```

```
//Khai báo
Button btnCapnhat;
int vitri = -1;
```

```
//ánh xạ
btnCapNhat = (Button) findViewById(R.id.btnCapNhat);
```

```
//bắt sự kiện trên từng dòng của ListView
listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
        //lấy nội dung đưa lên editText
        editText1.setText(arrayList.get(i));
        vitri = i;
    }
});
```

## ListView

Thêm môn học:

THÊM

CẬP NHẬT

Java

C#

PHP

```
btnCapNhat.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        arrayList.set(vitri, editText1.getText().toString());
        adapter.notifyDataSetChanged();
    }
});
```

```
//Xóa item
arrayList.remove(i);
adapter.notifyDataSetChanged();
```

Bước 1:  
Tạo Class  
MonHoc

Bước 3:  
ArrayList<MonHoc>

Bước 4:  
MonHocAdapter

Bước 2:  
row\_monhoc.xml

Bước 5:  
Đưa dữ liệu  
ra ListView



Java  
Java 1



C#  
C# 1



PHP  
P

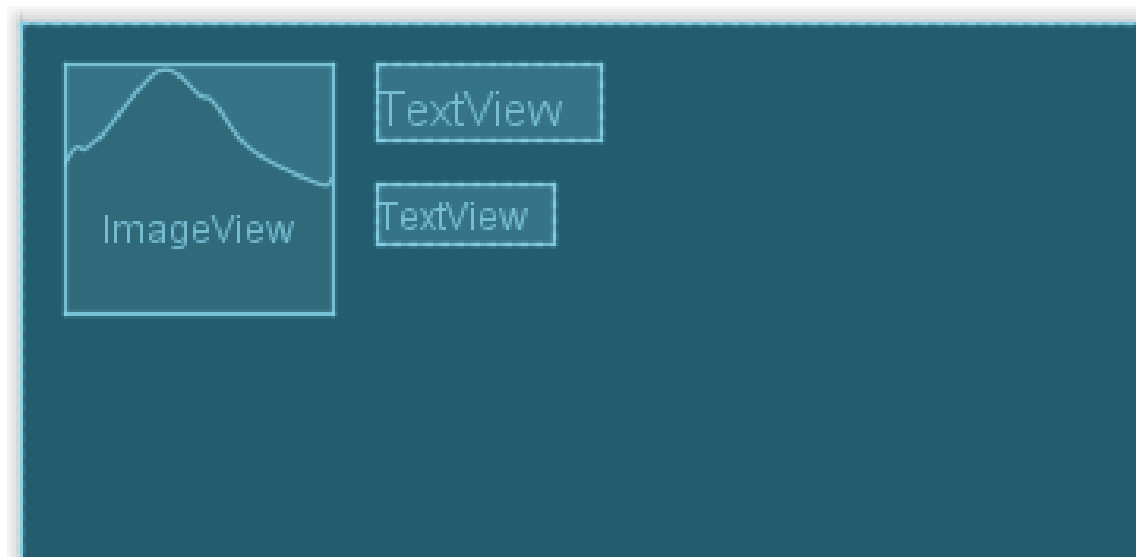
## Bước 1: Tạo Class MonHoc

```
package vn.iotstar.listview;

public class MonHoc {
    private String name;
    private String desc;
    private int pic;
    //kích phải chọn Generate.. Chọn
    constructor, getter and setter
```



Bước 2: tạo layout  
row\_monhoc.xml



## Bước 4: MonHocAdapter

```
//trả về số dòng
@Override
public int getCount() {
    return monHocList.size(); //lấy
    kích thước monhoclist
}
```

```
public class MonhocAdapter extends
BaseAdapter {
    //khai báo
    private Context context;
    private int layout;
    private List<MonHoc> monHocList;

    //tạo Constructors

    public MonhocAdapter(Context context,
int layout, List<MonHoc> monHocList) {
        this.context = context;
        this.layout = layout;
        this.monHocList = monHocList;
    }
```

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {
    //lấy context
    LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    //gọi view chứa layout
    view = inflater.inflate(layout,null);
    //ánh xạ view
    TextView textName = (TextView) view.findViewById(R.id.textName);
    TextView textDesc = (TextView) view.findViewById(R.id.textDesc);
    ImageView imagePic = (ImageView) view.findViewById(R.id.imagePic);

    //gán giá trị
    MonHoc monHoc = monHocList.get(i);
    textName.setText(monHoc.getName());
    textDesc.setText(monHoc.getDesc());
    imagePic.setImageResource(monHoc.getPic());
    //trả về view
    return view;
}
```



Java

Java 1



C#

C# 1



PHP

PHP 1

Bước 5:  
Đưa dữ liệu  
ra ListView

```
public class MainActivity extends AppCompatActivity {
    //khai báo
    ListView listView;
    ArrayList<MonHoc> arrayList;
    MonhocAdapter adapter;

    //ánh xạ
    AnhXa();

    //Tạo Adapter
    adapter = new MonhocAdapter(MainActivity.this,
        R.layout.row_monhoc,
        arrayList
    );
    //truyền dữ liệu từ adapter ra listview
    listView.setAdapter(adapter);
}
```



Java

Java 1

```
private void AnhXa() {
    listView = (ListView) findViewById(R.id.listview1);
```



C#

C# 1

```
    editText1 = (EditText) findViewById(R.id.editText1);
    btnNhap = (Button) findViewById(R.id.btnNhap);
    btnCapNhat = (Button) findViewById(R.id.btnCapNhat);
```



PHP

PHP 1

```
    //Thêm dữ liệu vào List
```

```
    arrayList = new ArrayList<>();
    arrayList.add(new MonHoc("Java","Java 1",R.drawable.java1));
    arrayList.add(new MonHoc("C#","C# 1",R.drawable.c));
    arrayList.add(new MonHoc("PHP","PHP 1",R.drawable.php));
    arrayList.add(new MonHoc("Kotlin","Kotlin
1",R.drawable.kotlin));
    arrayList.add(new MonHoc("Dart","Dart 1",R.drawable.dart));
}
```

**Bước 5:**  
Đưa dữ liệu  
ra ListView

*Mở MonHocAdapter lên*

*//tạo class viewholder*

```
private class ViewHolder{
    TextView textName,textDesc;
    ImageView imagePic;
}
```

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {
    //khởi tạo viewholder
    ViewHolder viewHolder;
    //lấy context
    if (view==null){
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        //gọi view chứa layout
        view = inflater.inflate(layout,null);
        //ánh xạ view
        viewHolder = new ViewHolder();
        viewHolder.textName = (TextView) view.findViewById(R.id.textName);
        viewHolder.textDesc = (TextView) view.findViewById(R.id.textDesc);
        viewHolder.imagePic = (ImageView) view.findViewById(R.id.imagePic);
        view.setTag(viewHolder);
    }else{
        viewHolder= (ViewHolder) view.getTag();
    }
}
```

*//gán giá trị*

```
MonHoc monHoc = monHocList.get(i);  
viewHolder.textName.setText(monHoc.getName());  
viewHolder.textDesc.setText(monHoc.getDesc());  
viewHolder.imagePic.setImageResource(monHoc.getPic());  
//trả về view  
return view;  
}
```





# KHOA CÔNG NGHỆ THÔNG TIN



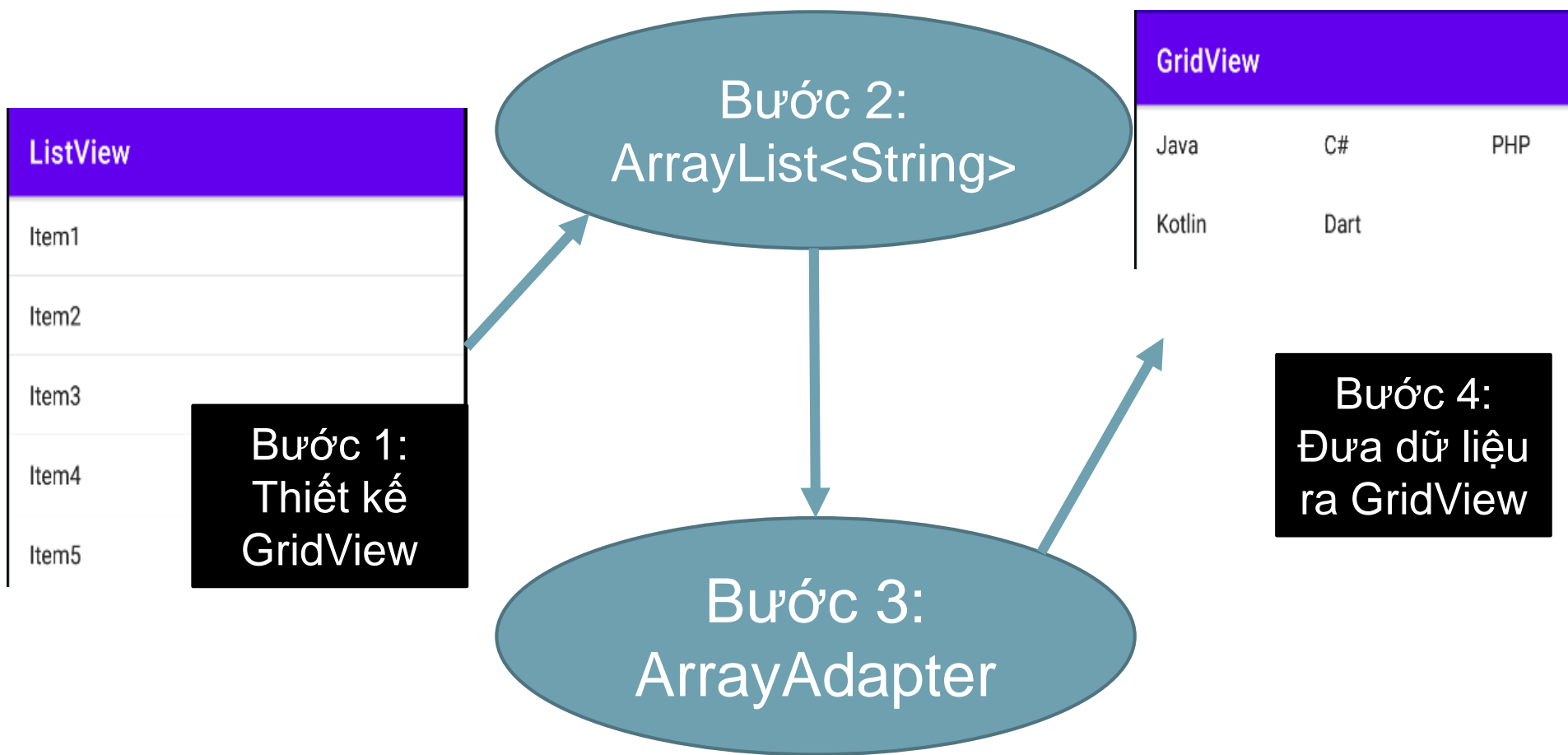
## GRIDVIEW

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



ThS. Nguyễn Hữu Trung

- GridView trong Android hiển thị các item trong mảng lưới hai chiều có thể scroll và các item này không cần thiết phải được định nghĩa trước, nhưng chúng tự động chèn vào Layout bởi sử dụng một ListAdapter.
- Các thuộc tính của GridView:
  - **android:verticalSpacing** -> Định nghĩa khoảng cách mặc định theo chiều dọc giữa các hàng. Có thể là trong px, dp, sp, in, hoặc mm.
  - **android:horizontalSpacing** -> Định nghĩa khoảng cách mặc định theo chiều ngang giữa các cột. Có thể là trong px, dp, sp, in, hoặc mm.
  - **android:numColumns** -> Xác định có bao nhiêu cột để hiển thị



Mở file .xml thiết kế GridView

GridView

Java	C#	PHP
Kotlin	Dart	

Bước 1:  
Thiết kế  
ListView

```
<GridView
    android:numColumns="3"
    android:id="@+id/gridview1"
    android:layout_width="409dp"
    android:layout_height="729dp"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp"
    tools:ignore="MissingConstraints" />
```

## Bước 2: ArrayList<String>

```
public class MainActivity extends
AppCompatActivity {
    //khai báo
    GridView gridView;
    ArrayList<String> arrayList;
```

*//ánh xạ*

```
gridView = (GridView) findViewById(R.id.gridview1);
```

*//Thêm dữ liệu vào List*

```
arrayList = new ArrayList<>();
```

```
arrayList.add("Java");
```

```
arrayList.add("C#");
```

```
arrayList.add("PHP");
```

```
arrayList.add("Kotlin");
```

```
arrayList.add("Dart");
```

## Bước 3: ArrayAdapter

- ArrayAdapter có 03 tham số:
- Context: màn hình hiển thị
  - Dạng Layout muốn đổ vào
  - Dữ liệu List đổ vào

```
//Tạo ArrayAdapter
```

```
ArrayAdapter adapter = new ArrayAdapter(  
    MainActivity.this,  
    android.R.layout.simple_list_item_1,  
    arrayList  
);
```

## GridView

Java	C#	PHP
Kotlin	Dart	

Bước 4:  
Đưa dữ liệu  
ra ListView

*//truyền dữ liệu từ adapter ra gridview*  
**gridView.setAdapter(adapter);**

## ListView

Java

C#

PHP

Kotlin

Dart

2

```
//bắt sự kiện click nhanh trên từng dòng của
GridView
gridView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?>
adapterView, View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên GridView -> i
        ban đầu =0
        Toast.makeText(MainActivity.this, "" + i,
        Toast.LENGTH_SHORT).show();
    }
});
```

Lấy nội dung: `arrayList.get(i)`



## ListView

Java

C#

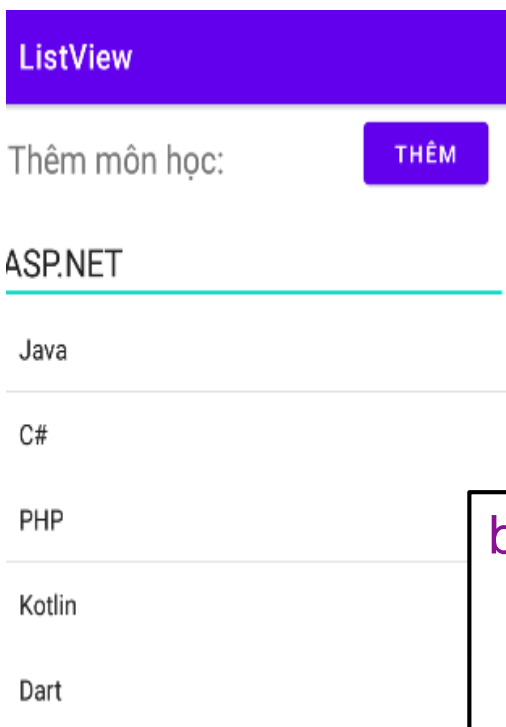
PHP

Kotlin

Dart

2

```
//bắt sự kiện click giữ trên từng dòng của Gridview
gridView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean
onItemLongClick(AdapterView<?> adapterView,
View view, int i, long l) {
        //code yêu cầu
        //i: trả về vị trí click chuột trên GridView -> i
        ban đầu =0
        Toast.makeText(MainActivity.this,"Bạn đang
nhấn giữ "+ i + "-" + arrayList.get(i) ,
Toast.LENGTH_SHORT).show();
        return false;
    }
});
```



```
EditText editText1;  
Button btnNhap;
```

```
editText1 = (EditText) findViewById(R.id.editText1);  
btnNhap = (Button) findViewById(R.id.btnNhap);
```

```
btnNhap.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String name = editText1.getText().toString();  
        arrayList.add(name);  
        adapter.notifyDataSetChanged();  
    }  
});
```

```
//Khai báo
Button btnCapnhat;
int vitri = -1;
```

```
//ánh xạ
btnCapNhat = (Button) findViewById(R.id.btnCapNhat);
```

```
//bắt sự kiện trên từng dòng của Gridview
gridView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view,
int i, long l) {
        //lấy nội dung đưa lên edittext
        editText1.setText(arrayList.get(i));
        vitri = i;
    }
});
```

ListView

Thêm môn học:

THÊM

CẬP NHẬT

Java

C#

PHP

```
btnCapNhat.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        arrayList.set(vitri, editText1.getText().toString());
        adapter.notifyDataSetChanged();
    }
});
```

```
//Xóa item
arrayList.remove(i);
adapter.notifyDataSetChanged();
```

Bước 1:  
Tạo Class  
MonHoc

Bước 3:  
ArrayList<MonHoc>

Bước 4:  
MonHocAdapter

Bước 2:  
row\_monhoc.xml

Bước 5:  
Đưa dữ liệu  
ra GridView

GridView



Java

Java 1



Kotlin

Kotlin 1



C#

C# 1



Dart



PHP

PHP 1



Dart 2

CN

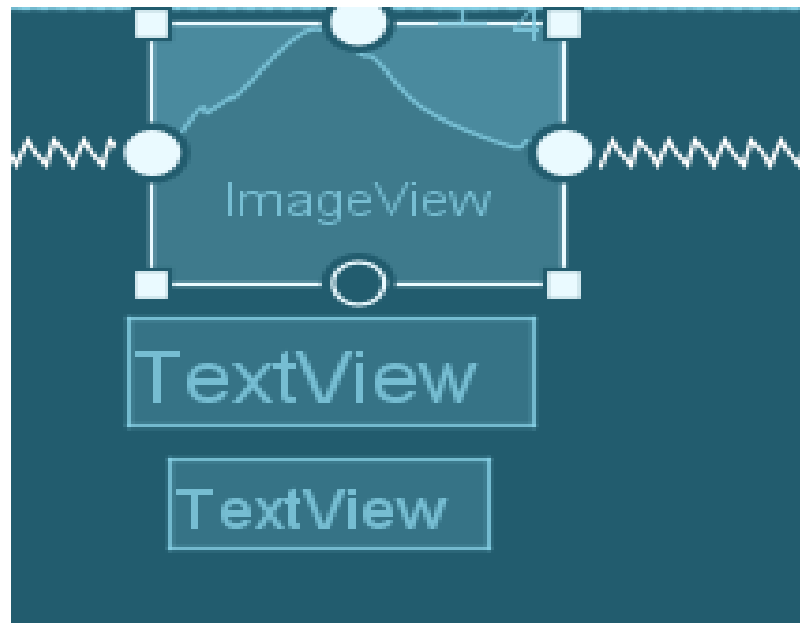
TT

## Bước 1: Tạo Class MonHoc

```
package vn.iotstar.gridview;

public class MonHoc {
    private String name;
    private String desc;
    private int pic;
    //kích phải chọn Generate.. Chọn
    constructor, getter and setter
```

Bước 2: tạo layout  
row\_monhoc.xml



## Bước 4: MonHocAdapter

```
//trả về số dòng
@Override
public int getCount() {
    return monHocList.size(); //lấy
    kích thước monhoclist
}
```

```
public class MonhocAdapter extends
BaseAdapter {
    //khai báo
    private Context context;
    private int layout;
    private List<MonHoc> monHocList;

    //tạo Constructors

    public MonhocAdapter(Context context,
int layout, List<MonHoc> monHocList) {
        this.context = context;
        this.layout = layout;
        this.monHocList = monHocList;
    }
```

```
monHocList == null ? 0 : monHocList.size()
```



@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {
    //lấy context
    LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    //gọi view chứa layout
    view = inflater.inflate(layout,null);
    //ánh xạ view
    TextView textName = (TextView) view.findViewById(R.id.textName);
    TextView textDesc = (TextView) view.findViewById(R.id.textDesc);
    ImageView imagePic = (ImageView) view.findViewById(R.id.imagePic);

    //gán giá trị
    MonHoc monHoc = monHocList.get(i);
    textName.setText(monHoc.getName());
    textDesc.setText(monHoc.getDesc());
    imagePic.setImageResource(monHoc.getPic());
    //trả về view
    return view;
}
```

## GridView

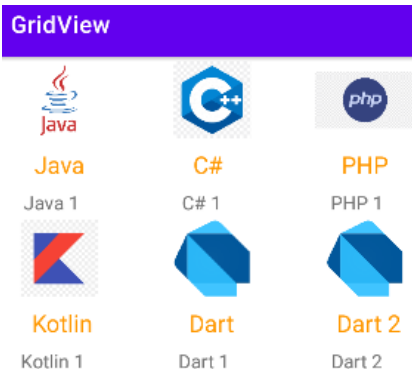


**Bước 5:**  
Đưa dữ liệu  
ra ListView

```
public class MainActivity extends AppCompatActivity {
    //khai báo
    ListView listView;
    ArrayList<MonHoc> arrayList;
    MonhocAdapter adapter;

    //ánh xạ
    AnhXa();

    //Tạo Adapter
    adapter = new MonhocAdapter(MainActivity.this,
        R.layout.row_monhoc,
        arrayList
    );
    //truyền dữ liệu từ adapter ra listview
    listView.setAdapter(adapter);
}
```



**Bước 5:**  
Đưa dữ liệu  
ra ListView

```
private void AnhXa() {
    gridView = (GridView) findViewById(R.id.gridview1);

    editText1 = (EditText) findViewById(R.id.editText1);
    btnNhap = (Button) findViewById(R.id.btnNhap);
    btnCapNhat = (Button) findViewById(R.id.btnCapNhat);

    //Thêm dữ liệu vào List
    arrayList = new ArrayList<>();
    arrayList.add(new MonHoc("Java","Java 1",R.drawable.java1));
    arrayList.add(new MonHoc("C#","C# 1",R.drawable.c));
    arrayList.add(new MonHoc("PHP","PHP 1",R.drawable.php));
    arrayList.add(new MonHoc("Kotlin","Kotlin
1",R.drawable.kotlin));
    arrayList.add(new MonHoc("Dart","Dart 1",R.drawable.dart));
}
```

*Mở MonHocAdapter lên*

*//tạo class viewholder*

```
private class ViewHolder{
    TextView textName,textDesc;
    ImageView imagePic;
}
```

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {
    //khởi tạo viewholder
    ViewHolder viewHolder;
    //lấy context
    if (view==null){
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        //gọi view chứa layout
        view = inflater.inflate(layout,null);
        //ánh xạ view
        viewHolder = new ViewHolder();
        viewHolder.textName = (TextView) view.findViewById(R.id.textName);
        viewHolder.textDesc = (TextView) view.findViewById(R.id.textDesc);
        viewHolder.imagePic = (ImageView) view.findViewById(R.id.imagePic);
        view.setTag(viewHolder);
    }else{
        viewHolder= (ViewHolder) view.getTag();
    }
}
```

*//gán giá trị*

```
MonHoc monHoc = monHocList.get(i);
viewHolder.textName.setText(monHoc.getName());
viewHolder.textDesc.setText(monHoc.getDesc());
viewHolder.imagePic.setImageResource(monHoc.getPic());
```

*//trả về view*

```
return view;
```

```
}
```

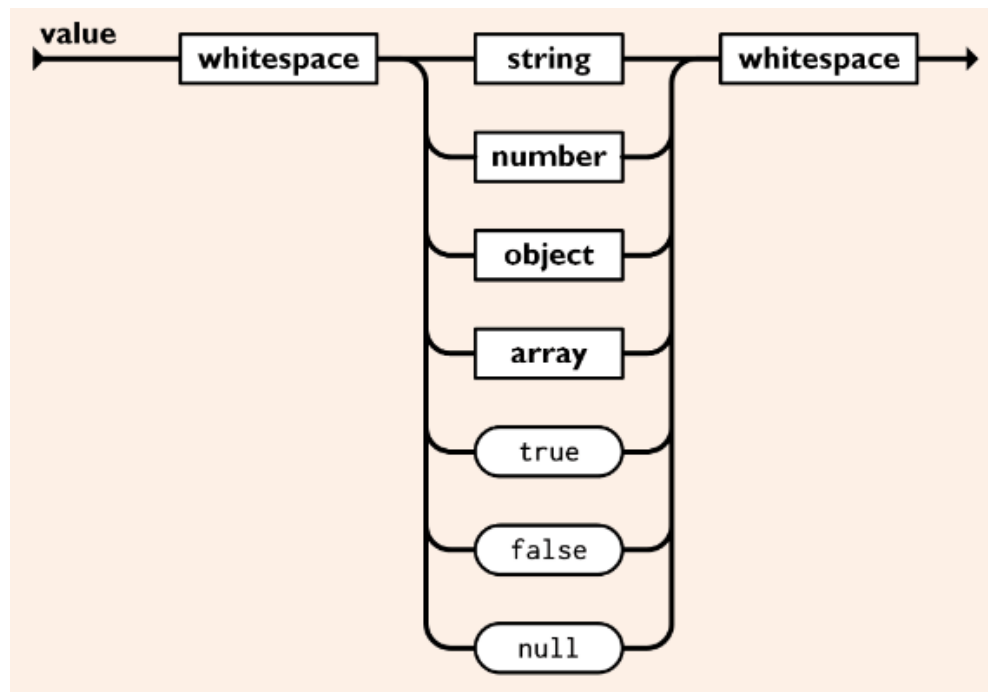
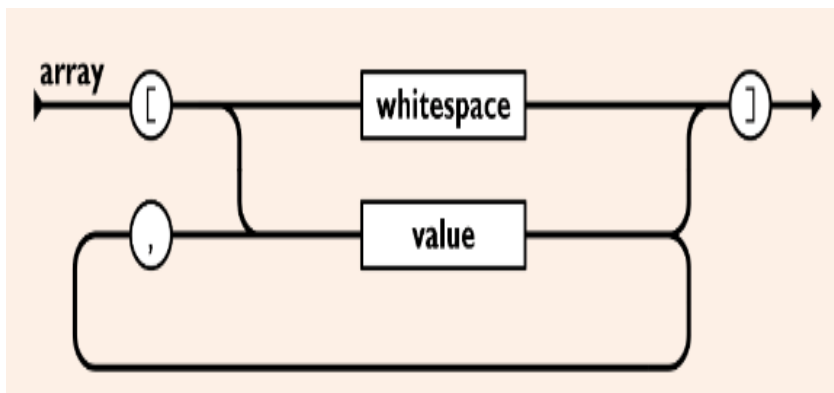
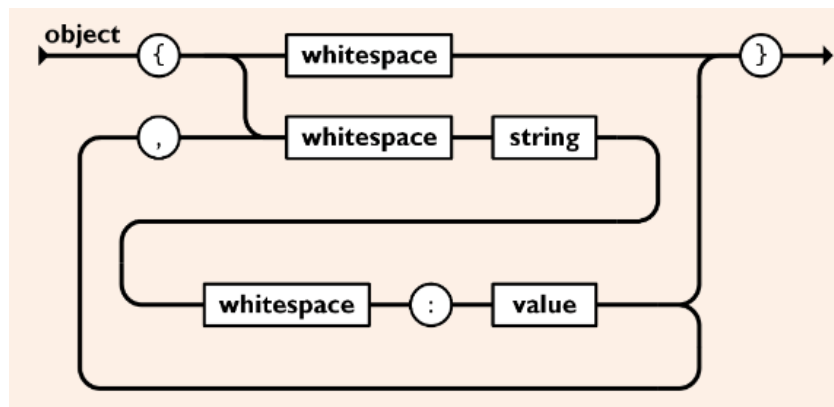


## Kết nối Android với JSON

Khoa Công nghệ Thông tin  
Đại học Sư phạm Kỹ thuật TP.HCM



- Truy cập trang [json.org](http://json.org) xem cấu trúc JSON (Javascript Object Notation)





- Tìm json formatter & Validator

## JSON Data/URL

```
{ "name" : "Java",  
  "desc" : "Java 1" }
```

## Formatted JSON Data

```
{  
  "name" : "Java",  
  "desc" : "Java 1"  
}
```

## □ Tải file data.json về bỏ vào web

← → ↻ ⚠ Not secure | 192.168.31.214:8090/data.json

Android - Login Scr... Kotlin Tutorial Setting Up Swagger...

```
{
  "name": "Java",
  "desc": "Java 1"
}
```

## Network Connection Details

### Network Connection Details:

Property	Value
Connection-specific DNS S...	
Description	Realtek RTL8188EU Wireless LAN 802.11n U
Physical Address	50-3E-AA-A3-2C-61
DHCP Enabled	Yes
IPv4 Address	192.168.31.214
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Friday, January 13, 2023 6:45:45 PM
Lease Expires	Saturday, January 14, 2023 6:45:39 AM
IPv4 Default Gateway	192.168.31.1
IPv4 DHCP Server	192.168.31.1
IPv4 DNS Server	192.168.31.1
IPv4 WINS Server	
NetBIOS over Tcpip Enabl...	Yes
Link-local IPv6 Address	fe80::5b2d:3aef:36cc:2856%41
IPv6 Default Gateway	
IPv6 DNS Server	

Close

- Khởi tạo dự án Empty Android. Tạo Class ReadJSON

*//Tạo Class đọc Json*

```
private class ReadJSONObject extends AsyncTask<String,Void,String>{
    @Override
    protected String doInBackground(String... strings) {
        StringBuilder content = new StringBuilder();
        try {
            URL url = new URL(strings[0]);
            InputStreamReader inputStreamReader = new
            InputStreamReader(url.openConnection().getInputStream());
            BufferedReader bufferedReader = new BufferedReader(inputStreamReader);
            String line = "";
            while ((line = bufferedReader.readLine()) != null){
                content.append(line);
            }
            bufferedReader.close();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return content.toString();
    }
}
```

@Override

```
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    //phân tích JSON
    try {
        JSONObject object = new JSONObject(s);
        String name = object.getString("name");
        String desc = object.getString("desc");
        String pic = object.getString("pic");

        String kq = name + "\n" + desc + "\n" + pic;
        Toast.makeText(MainActivity.this, kq, Toast.LENGTH_SHORT).show();

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

## JSON Data/URL

```
{
  "monhoc" : [
    {
      "name" : "Java", "desc" : "Java 1", "pic" : "pic1.png"},
    {
      "name" : "C#", "desc" : "C# 1", "pic" : "pic2.png"},
    {
      "name" : "Kotlin", "desc" : "Kotlin 1", "pic" : "pic3.png"},
    {
      "name" : "PHP", "desc" : "PHP 1", "pic" : "pic4.png"},
    {
      "name" : "Dart", "desc" : "Dart 1", "pic" : "pic5.png"}
  ]
}
```

## JSON Data/URL

```
{
  "monhoc" : [
    {
      "name" : "Java", "desc" : "Java 1", "pic" : "pic1.png"},
    {
      "name" : "C#", "desc" : "C# 1", "pic" : "pic2.png"},
    {
      "name" : "Kotlin", "desc" : "Kotlin 1", "pic" : "pic3.png"},
    {
      "name" : "PHP", "desc" : "PHP 1", "pic" : "pic4.png"},
    {
      "name" : "Dart", "desc" : "Dart 1", "pic" : "pic5.png"}
  ]
}
```

```
new ReadJSONObject().execute("http://192.168.31.214:8090/data3.json");
```

@Override

```
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    //phân tích JSON
    try {
        JSONObject object = new JSONObject(s);
        //xử lý mảng
        JSONArray array = object.getJSONArray("monhoc");
        //duyet các phần tử trong mảng
        for(int i=0;i<array.length();i++){
            JSONObject object1 = array.getJSONObject(i);
            String name = object1.getString("name");
            String desc = object1.getString("desc");
            String pic = object1.getString("pic");
            String kq = name + "\n" + desc + "\n" + pic;
            Toast.makeText(MainActivity.this,kq,Toast.LENGTH_SHORT).show();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

@Override

```
protected void onPostExecute(String s) {
    super.onPostExecute(s);
    //phân tích JSON
    try {
        JSONObject object = new JSONObject(s);
        JSONObject objectLang = object.getJSONObject("language");
        JSONObject objectVN = objectLang.getJSONObject("vn");
        String ten = objectVN.getString("ten");

        Toast.makeText(MainActivity.this,ten,Toast.LENGTH_SHORT).show();
    }
    catch (JSONException e) {
        e.printStackTrace();    }    }
```



- Nguyễn Hữu Trung
- 0908617108
- [trungnh@hcmute.edu.vn](mailto:trungnh@hcmute.edu.vn)
- [utex.hcmute.edu.vn](http://utex.hcmute.edu.vn)