



Lập trình Android

Bài 4. *Tài nguyên ứng dụng cơ bản*

Ngành Mạng & Thiết bị di động





Nội dung

1. Khái niệm

- Tài nguyên & Tính tương thích
- Định nghĩa tài nguyên
- Truy xuất tài nguyên
- Tài nguyên Alias

2. Các tài nguyên cơ bản



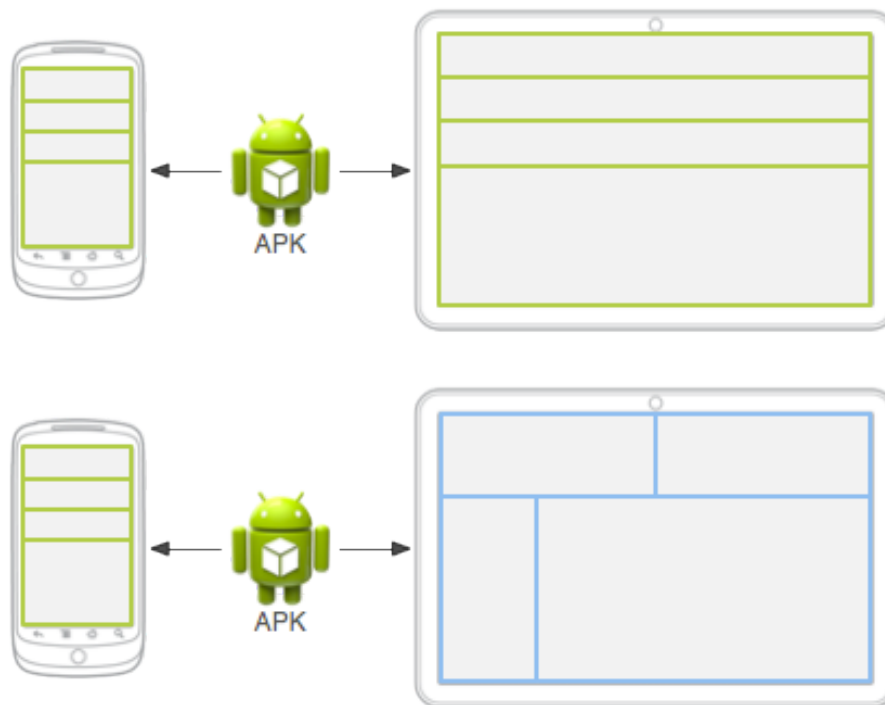
1.1 Tài nguyên và tính tương thích

- ❑ Tài nguyên: một dạng dữ liệu được xây dựng nhằm đáp ứng các yêu cầu về hiển thị bao gồm hình ảnh, âm thanh, văn bản, các bố cục...tương thích cho từng thiết bị riêng biệt.
- ❑ Cho phép khai báo một lần và sử dụng trong phạm vi toàn ứng dụng, dễ dàng thay đổi theo ngữ cảnh.
- ❑ Tính tương thích: để có thể tối ưu hóa tính tương thích thiết bị tài nguyên được chia làm hai dạng:
 - Tài nguyên mặc định: không quan tâm đến cấu hình của thiết bị hoặc không có tài nguyên để lựa chọn.
 - Tài nguyên đặc trưng: được sử dụng trên thiết bị riêng biệt thông qua các từ hạn định và đường dẫn.



1.1 Tài nguyên và tính tương thích

- ❑ Ứng dụng tự lựa chọn tài nguyên phù hợp với thông tin cấu hình thiết bị, tài nguyên mặc định được chọn nếu không có tài nguyên phù hợp.





1.2 Định nghĩa tài nguyên

❑ Tài nguyên ứng dụng được định nghĩa trong thư mục **res** của dự án, bao gồm các dạng tài nguyên sau:

- Animator
- Anim
- Color
- Drawable
- Layout
- Menu
- Raw
- Values
- XML



1.2 Định nghĩa tài nguyên

❑ Vấn đề về định nghĩa tài nguyên:

- Có quá nhiều thiết bị có cấu hình khác nhau về kích thước màn hình, độ phân giải, phím vật lý...
- Mỗi thiết bị có thể hoạt động ở nhiều chế độ khác nhau: nằm ngang, nằm đứng, thay đổi ngôn ngữ...

❑ Từ hạn định: dùng để tạo ra các tài nguyên khác nhau cho nhiều thiết bị có cấu hình khác nhau hoạt động ở các chế độ khác nhau.

- Ví dụ:

drawable

ic_launcher.png

drawable-hdpi

ic_launcher.png



1.2 Định nghĩa tài nguyên

❑ Các dạng từ hạn định:

- MCC – MNC
 - `mcc452-mnc04`
- Language & Region
 - `vi-rVN`
- Layout Direction (API 17)
 - `ldltr - ldrtl`
- Smallest Width
 - `sw320dp – sw480dp – sw600dp - sw720dp`
- Available Width
 - `w720dp – w1024dp`
- Available Height
 - `h720dp – h1024dp`



1.2 Định nghĩa tài nguyên

☐ Các dạng từ hạn định:

- Screen Size
 - small – normal – large - xlarge
- Screen Aspect
 - long - notlong
- Screen Orientation
 - port- land
- UI Mode
 - car – desk – television – appliance
- Night Mode
 - night – notnight
- Screen pixel density
 - ldpi – mdpi – hdpi – xhdpi – nodpi - tvdpi



1.2 Định nghĩa tài nguyên

☐ Các dạng từ hạn định:

- Touch Screen
 - notouch - finger
- Keyboard
 - keysexposed – keyshidden - keyssoft
- Input Method
 - nokeys – qwerty – 12key
- Navigation Key
 - Navexposed – navhidden
- Non-Touch Navigation
 - nonav – dpad – trackball- wheel
- Platform Version
 - v3 – v4 – v7 – v11...



1.2 Định nghĩa tài nguyên

❑ Cách tạo tài nguyên:

- Tạo thư mục mới trong thư mục **res** với định dạng:
 <tên tài nguyên>-<từ hạn định>
- Ví dụ:
 drawable-vi-rVN

❑ Quy tắc đặt tên cho thư mục tài nguyên:

- Có thể có nhiều từ hạn định cho một thư mục tài nguyên cách nhau bằng dấu gạch ngang (“-”).
- Các từ hạn định phải theo thứ tự ưu tiên.
- Các thư mục tài nguyên không được chứa thư mục tài nguyên khác.
- Không cho phép hai từ hạn định giống nhau trên cùng một thư mục.



1.3 Truy xuất tài nguyên

- ❑ Tất cả tài nguyên ứng dụng được truy xuất thông qua lớp R.
- ❑ Lớp R:
 - Lớp tĩnh.
 - Chứa trong thư mục gen, tự động tạo các định danh cho tài nguyên (ID) thông qua AAPT (Android Application Project Tool).
 - Chứa các lớp tài nguyên, mỗi dạng tài nguyên là một lớp tĩnh.
 - Ví dụ:
 - Truy xuất tài nguyên hình ảnh:
 - Java code: `R.drawable.ic_launcher`
 - XML: `@drawable/ic_launcher`



1.3 Truy xuất tài nguyên

❑ Cú pháp dùng chung khi truy xuất:

Java Code:

`[<package_name.>]R.<resource_type>.<resource_name>`

XML:

`@[<package_name.>:]<resource_type>/<resource_name>`

Trong đó:

- `Package_name`: tên gói ứng dụng
- `Resource_type`: dạng tài nguyên
- `Resource_name`:
 - Tên tài nguyên cần truy xuất không bao gồm phần mở rộng tập tin
 - Thuộc tính **android:name** dành cho các tài nguyên cơ bản (string, color...).



1.4 Tài nguyên Alias

- ❑ Cho phép tạo ra tài nguyên từ tài nguyên có sẵn, phục vụ cho nhiều cấu hình thiết bị nhưng không phải là tài nguyên mặc định.
- ❑ Ví dụ:
 - Vấn đề: tạo biểu tượng ứng dụng khác nhau cho các ngôn ngữ khác nhau, đối với tiếng Anh và tiếng Việt thì cùng biểu tượng.
 - Giải quyết vấn đề (không dùng Alias):
 - Tạo thư mục tài nguyên cho từng ngôn ngữ.
 - Chép hình ảnh khác nhau cho từng thư mục, hai thư mục có từ hạn định **en** và **vi** có hình ảnh giống nhau.
 - Giải quyết vấn đề (dùng Alias):
 - Tạo thư mục tài nguyên cho từng ngôn ngữ.
 - Chép hình ảnh khác nhau cho từng thư mục.
 - Thư mục có từ hạn định **en** tạo tài nguyên Alias từ thư mục **vi**.



Nội dung

1. Khái niệm

2. Các tài nguyên cơ bản

- String
- Color
- Dimen
- Array



2. Các tài nguyên cơ bản

- ❑ Các tài nguyên cơ bản được lưu trữ trong thư mục res/**values**.
- ❑ Định danh tài nguyên được khởi tạo thông qua thuộc tính **name**, không phải tên tập tin.
- ❑ Có thể lưu trữ nhiều tài nguyên vào trong một tập tin.
- ❑ Một số tên tập tin đề xuất trong values:
 - string.xml
 - arrays.xml
 - colors.xml
 - dims.xml
- ❑ Tất cả các tập tin xml trong values, được mở đầu và kết thúc bằng cặp thẻ **<resource></resource>**.





2.1 String

- ❑ Cung cấp tài nguyên dạng văn bản cho ứng dụng, cho phép thực hiện các thao tác định dạng và thiết kế khác nhau, bao gồm ba dạng:
 - String
 - StringArray
 - QuantityString (Plural)



2.1 String

❑ String:

- Khai báo:

```
<string name="string_name">Text_string</string>
```

Trong đó:

- string_name: định danh dùng để truy xuất trong XML và Java Code.
 - text_string: nội dung lưu trữ.
- Ví dụ: tập tin **strings.xml**

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

```
<resource>
```

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

```
</resource>
```



2.1 String

□ String:

- Ví dụ truy xuất và sử dụng: Khai báo TextView và gắn văn bản cho thuộc tính text.

- Truy xuất trong XML:

```
<TextView
```

```
...
```

```
    android:text="@string/hello" />
```

- Truy xuất trong Java Code:

Truy xuất trực tiếp:

```
textView.setText(R.string.hello);
```

Dùng phương thức getString:

```
textView.setText(getString(R.string.hello));
```



2.1 String

❑ **StringArray:**

- Khai báo:

```
<string-array name="string_array_name">  
    <item>Text_string</item>  
</string-array>
```

Trong đó:

- `string_array_name`: định danh dùng để truy xuất trong XML và Java Code.
- `text_string`: nội dung lưu trữ cho từng item.

- Ví dụ:

```
<string-array name="android_courses">  
    <item>Basic Android</item>  
    <item>Advanced Android</item>  
</string-array>
```



2.1 String

❑ **StringArray:**

- Truy xuất trong Java Code:

```
Resources res = getResources();
```

```
String[] androidCourses =
```

```
    res.getStringArray(R.array.android_courses);
```



2.1 String

- ❑ **Quantity:** được sử dụng cùng bộ số đếm tùy thuộc vào qui ước của từng ngôn ngữ, bao gồm các bộ đếm:
 - Zero
 - One
 - Two
 - Few
 - Many
 - Other

- ❑ Thường được dùng kết hợp với các định dạng số nguyên.

- ❑ Hệ thống tự động lựa chọn tùy chọn bộ đếm tùy thuộc vào số đếm và ngôn ngữ sử dụng trên thiết bị.



2.1 String

❑ Quantity:

- Khai báo:

```
<plurals name="plural_name">
```

```
    <item quantity="count">Text_string</item>
```

```
</plurals>
```

Trong đó:

- plural_name: định danh dùng để truy xuất trong XML và Java Code.
- count: bộ đếm sử dụng.
- text_string: nội dung lưu trữ cho từng item.



2.1 String

❑ Quantity:

- Ví dụ:

```
<plurals name="numberOfBooks">
```

```
    <item quantity="one">One book found.</item>
```

```
    <item quantity="other">%d books found.</item>
```

```
</plurals>
```

- Truy xuất trong Java Code:

```
Resources res = getResources();
```

```
String booksFound =
```

```
    res.getQuantityString(R.plurals.numberOfBooks, 2, 2);
```



2.1 String

❑ Định dạng String:

- Truyền tham số: cho phép thực hiện tạo các đoạn văn bản có chứa tham số truyền vào.
- Ví dụ:

```
<string name="messages">
```

Chào %1\$s! Bạn có %2\$d tin nhắn mới.

```
</string>
```

- Truy xuất trong Java Code:

```
String messages = getString(R.string.messages);  
messages = String.format(messages, "HTSI", 10);  
Log.d("HTSI", messages);  
// Chào HTSI! Bạn có 10 tin nhắn mới.
```




2.1 String

❑ Định dạng String:

- Định dạng HTML: cho phép hiển thị các định dạng cấu trúc HTML.
- Ví dụ:

```
<string name="messages">
```

Chào %1\$s! Bạn có %2\$d tin nhắn mới .

```
</string>
```

- Truy xuất trong Java Code:

```
String messages = getString(R.string.messages);
```

```
messages = String.format(messages, "HTSI", 10);
```

```
CharSequence htmlMessages = Html.fromHtml(messages);
```

```
Log.d("HTSI", htmlMessages);
```

```
// Chào HTSI! Bạn có 10 tin nhắn mới.
```



2.2 Color

- ❑ Khai báo tài nguyên sử dụng cho các hiển thị màu sắc như phông nền, hình ảnh, màu chữ...

- ❑ Có thể sử dụng các định dạng màu sắc sau:
 - #RGB
 - #ARGB
 - #RRGGBB
 - #AARRGGBB



2.2 Color

❑ Khai báo:

```
<color name="color_name">hex_color</color>
```

Trong đó:

- color_name: định danh dùng để truy xuất trong XML và Java Code.
- hex_color: định dạng màu sắc

● Ví dụ: tập tin **colors.xml**

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

```
<resource>
```

```
    <color name="my_blue">#33B5E5</color>
```

```
    <color name="my_green">#99CC00</color>
```

```
</resource>
```



2.2 Color

- ❑ Ví dụ truy xuất và sử dụng: Khai báo TextView và thiết lập màu văn bản thông qua thuộc tính textColor.

- Truy xuất trong XML:

```
<TextView
```

```
...
```

```
    android:textColor="@color/my_blue" />
```

- Truy xuất trong Java Code:

Dùng phương thức getColor:

```
Resources res = getResources();
```

```
int myBlue = res.getColor(R.color.my_bule);
```

```
textView.setTextColor(myBlue);
```



2.2 Color

- ❑ **ColorStateList**: đối tượng cho phép xây dựng một tập các màu sắc khác nhau hiển thị cho các trạng thái khác nhau:
 - Pressed
 - Focused
 - Selected
 - Checkable
 - Checked
 - Enable
 - Window_focused
- ❑ Khai báo trong thư mục **res/color**, tập tin XML khai báo trong thư mục này bắt đầu và kết thúc bằng cặp thẻ **<selector></selector>**.
- ❑ Định danh **ColorStateList** được truy xuất thông tên tập tin (không bao gồm phần mở rộng).





2.2 Color

❑ ColorStateList:

- Khai báo:

<selector

xmlns:android="http://schemas.android.com/apk/res/android">

<item [android:state = "true|false"]
 android:color="hex_color" />

</selector>

Trong đó:

- state: chỉ định trạng thái thông qua **true** hoặc **false**, nếu không có chỉ định trạng thái là bình thường.
- hexa_color: màu tương ứng với trạng thái.



2.2 Color

❑ ColorStateList:

- Ví dụ: tập tin **button_text_selector.xml** trong **res/color**

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">  
    <item android:state_pressed="true" android:color="@color/my_green"/>  
    <item android:color="@color/my_blue"/>  
</selector>
```

- Truy xuất sử dụng trong XML

```
<button  
    ...  
    android:textColor="@color/button_text_selector"  
/>
```



2.3 Dimen

- ❑ Khai báo tài nguyên sử dụng cho các đại lượng kích thước trong ứng dụng.

- ❑ Có thể sử dụng các đại lượng kích thước sau:
 - dp – dip
 - sp
 - pt
 - Px
 - mm
 - in



2.3 Dimen

❑ Khai báo:

```
<dimen name="dimen_name">size</dimen>
```

Trong đó:

- `dimen_name`: định danh dùng để truy xuất trong XML và Java Code.
 - `size`: đại lượng đi kèm định dạng
- Ví dụ: tập tin **dimens.xml**

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

```
<resource>
```

```
    <dimen name="text_size">20sp</dimen>
```

```
</resource>
```



2.3 Dimen

- ❑ Ví dụ truy xuất và sử dụng: Khai báo Button và thiết lập kích thước văn bản thông qua thuộc tính `textSize`.

- Truy xuất trong XML:

```
<TextView
```

```
...
```

```
    android:textSize="@dimen/text_size" />
```

- Truy xuất trong Java Code:

Dùng phương thức `getDimension`:

```
Resources res = getResources();
```

```
float textSize = res.getDimension(R.dimen.text_size);
```

```
textView.setTextSize(textSize);
```



2.4 TypedArray

- ❑ Khai báo tài nguyên cho phép xây dựng tập hợp (mảng) lưu trữ các loại tài nguyên như hình ảnh, màu sắc... hoặc có thể lưu trữ cùng lúc nhiều dạng tài nguyên khác nhau.
- ❑ Khai báo:

```
<array name="array_name">
```

```
    <item>value</item>
```

```
</array>
```

Trong đó:

- array_name: định danh dùng để truy xuất trong XML và Java Code.
- value: giá trị lưu trữ



2.4 TypedArray

❑ Ví dụ: tập tin **arrays.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resource>
    <array name="colors">
        <item>@color/my_blue</item>
        <item>@color/my_green</item>
        <item>#FFBB33</item>
    </array>
</resource>
```

Truy xuất và sử dụng trong Java Code:

```
TypedArray colors = res.obtainTypedArray(R.array.colors);
int myBlue = colors.getColor(2,0);
```

