

Bài 06:

TÀI NGUYÊN ỨNG DỤNG TRONG

ANDROID

GVGD: ThS. Đặng Thế Hân

Biên soạn: ThS. Giang Hào Côn

Mục tiêu

Cung cấp cho sinh viên kiến thức cơ bản về đa ngôn ngữ trong ứng dụng Android và cách xử lý tập tin ngôn ngữ lập trình ứng dụng trên Android.

Nội dung

- 1) Đa ngôn ngữ
- 2) Xử lý tập tin trong Android

5.1/ Đa ngôn ngữ

- **Đa ngôn ngữ trong android** giúp cho ứng dụng android có thể hiển thị giao diện với nhiều ngôn ngữ khác nhau tùy thuộc vào việc người dùng lựa chọn ngôn ngữ hiển thị trên thiết bị của họ.
- ▪ Mặc định android sử dụng ngôn ngữ tiếng Anh là ngôn ngữ chính và nạp chuỗi từ **res** \Rightarrow **values** \Rightarrow **strings.xml**. Do đó, nếu ta muốn bổ sung thêm những ngôn ngữ khác, ta cần phải tạo thư mục với tên được đặt theo quy tắc là **value-mã ngôn ngữ** theo từng quốc gia.
- ▪ Ví dụ nếu ta muốn thêm tiếng Việt, bạn sẽ phải tạo một thư mục với tên **values-vi**, sao chép tập tin **strings.xml** vào trong thư mục này và dịch tất cả các chuỗi sang **ngôn ngữ tiếng Việt**.

5.1/ Đa ngôn ngữ

▪ Bảng Ngôn ngữ Android hỗ trợ.

Language	Locale	values/strings.xml
German	de	values-de/strings.xml
Chinese	zh	values-zh/strings.xml
Czech	cs	values-cs/strings.xml
Dutch	nl	values-nl/strings.xml
French	fr	values-fr/strings.xml
Japanese	ja	values-ja/strings.xml
Korean	ko	values-ko/strings.xml
Thai	th	values-th/strings.xml
Vietnamese	vi	values-vi/strings.xml

5.1/ Đa ngôn ngữ

Hoạt động đa Ngôn ngữ Android như sau:

- Khi người dùng thay đổi ngôn ngữ điện thoại thông qua thiết lập **Settings** \Rightarrow **Language & Input**, hệ điều hành Android sẽ tự kiểm tra đối với các nguồn tài nguyên ngôn ngữ thích hợp trong ứng dụng.
- Nếu ứng dụng hỗ trợ lựa chọn ngôn ngữ, android tìm kiếm các nguồn tài nguyên phù hợp với ngôn ngữ được chọn từ **values-(mã ngôn ngữ)**.
- Nếu các chuỗi bị thiếu, android sẽ nạp các chuỗi đó từ tập tin **strings.xml** (**res** \Rightarrow **values** \Rightarrow **strings.xml**)

5.1/ Đa ngôn ngữ

Một số lưu ý khi xử lý đa ngôn ngữ:

- Luôn luôn khai báo các chuỗi trong tập tin **strings.xml**
`<string name="your_address">Enter your address</string>`
- Sử dụng nó trong tập tin giao diện xml
`<TextView ... android:text="@string/your_address" />`
- Truy xuất nó trong mã java
`txtAddress.setText(R.string.your_address);`

5.1/ Đa ngôn ngữ

Một số lưu ý khi xử lý đa ngôn ngữ:

- **Không nên làm chỉ định trực tiếp trong xml hay trong mã java**

```
<TextView ... android:text="Enter your address" />
```

```
txtAddress.setText("Enter your address");
```

- **Các bước thực hiện**

Sử dụng project hiện có hoặc tạo mới project trong Android Studio và thực hiện các bước bên dưới để tạo ứng dụng android hỗ trợ 2 ngôn ngữ là tiếng Việt và tiếng Anh

5.1/ Đa ngôn ngữ

Các bước thực hiện:

Bước 1: Mở tập tin strings.xml và bổ sung nội dung sau trước thẻ **</resources>**.

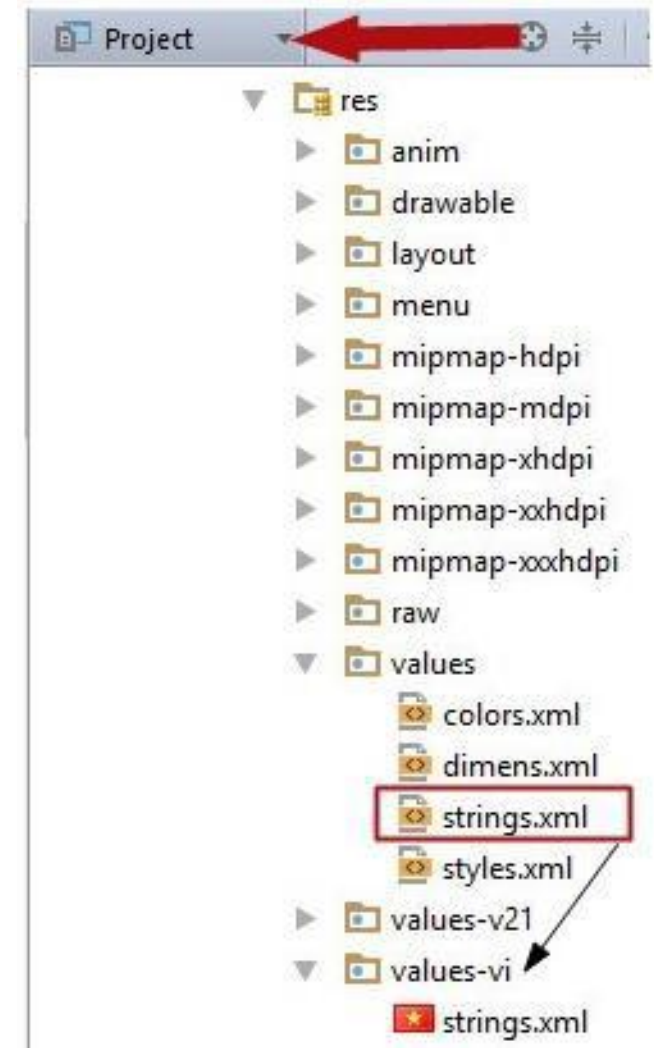
```
<string name="welcome">Welcome!</string>
<string name="username">Username</string>
<string name="password">Password</string>
<string name="login">Login</string>
<string name="signup">Don\'t have account? Sign Up</string>
```

5.1/ Đa ngôn ngữ

Các bước thực hiện:

Bước 2: Tạo thư mục với tên values-vi bên dưới thư mục res, thực hiện: Chuột phải **res** => chọn **New** => chọn **Directory** => nhập tên thư mục là **values-vi**.

Bước 3: Sao chép tập tin **strings.xml** từ thư mục **values** và dán vào thư mục **values-vi**.



5.1/ Đa ngôn ngữ

Các bước thực hiện:

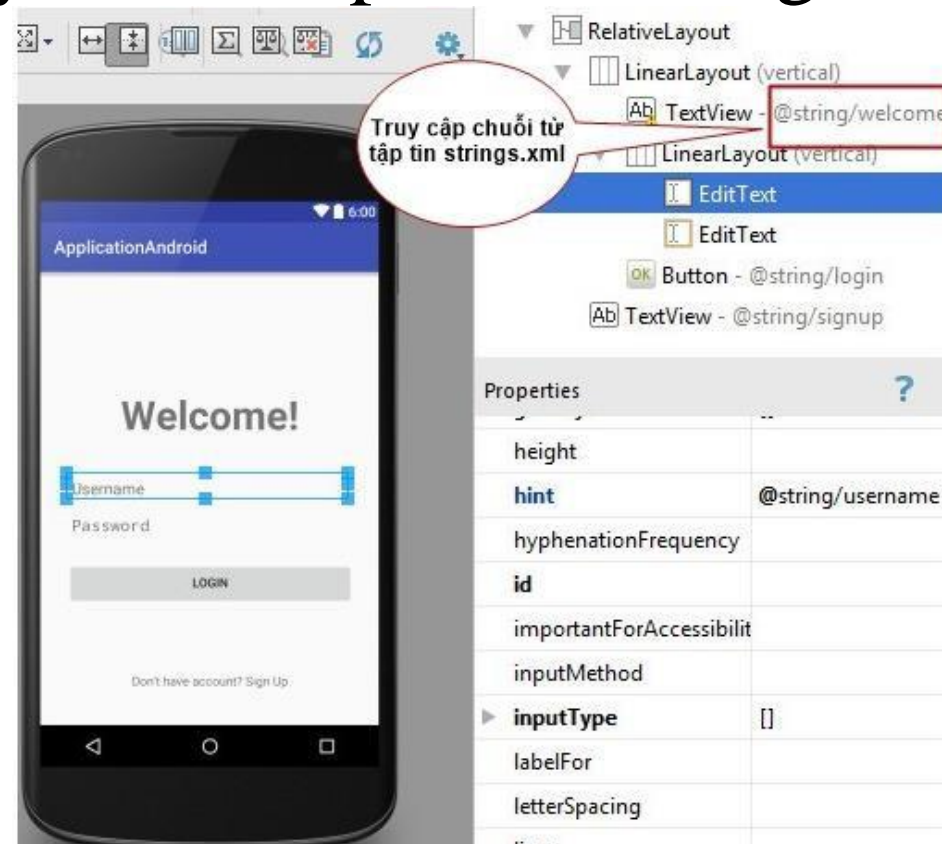
Bước 4: Viết hoá nội dung trong tập tin **strings.xml** của thư mục **values-vi**.

```
<string name="welcome">Xin chào!</string>
<string name="username">Tên đăng nhập</string>
<string name="password">Mật khẩu</string>
<string name="signup">Chưa có tài khoản? Đăng ký</string>
<string name="login">Đăng nhập</string>
```

5.1/ Đa ngôn ngữ

Các bước thực hiện:

Bước 5: Tạo activity tên SampleMultiLangActivity và thiết kế giao diện sau:.

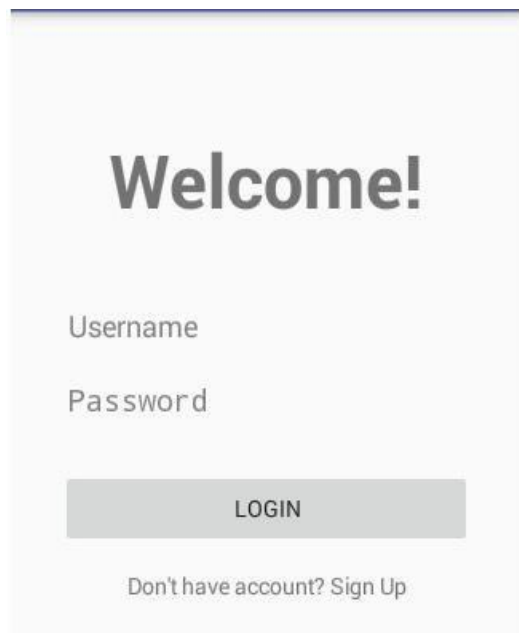


5.1/ Đa ngôn ngữ

Các bước thực hiện:

Bước 6: Chạy ứng dụng.

Ngôn ngữ thiết bị là English



Wellcome!

Username

Password

LOGIN

Don't have account? Sign Up

Thay đổi ngôn ngữ thiết bị
bằng cách chọn **Setting** =>
chọn Language & Input =>
chọn Tiếng Việt



Xin chào!

Tên đăng nhập

Mật khẩu

ĐĂNG NHẬP

Chưa có tài khoản? Đăng ký

5.2/ Xử lý tập tin

- Xử lý tập tin trong Android thường có 2 công việc là Lưu trữ và
- đọc dữ liệu.
- Lưu trữ dữ liệu trong android cung cấp nhiều lựa chọn trong việc lưu trữ dữ liệu như:
 - Bộ nhớ trong (internal storage)
 - Lưu trữ dữ liệu vào bộ nhớ ngoài (external storage)
 - Lưu trữ dữ liệu với shared preferences
 - Lưu trữ dữ liệu sử dụng sqlite.

5.2/ Xử lý tập tin

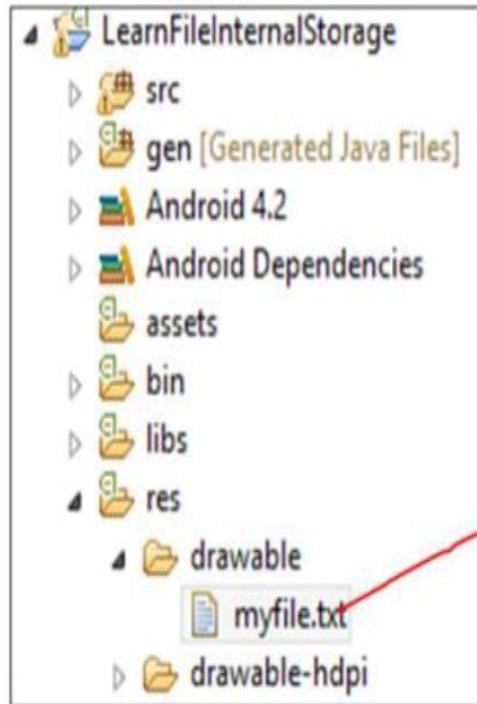
Bộ nhớ trong (Internal Storage) – Đọc tập tin trong Drawable.

Giả sử tạo 1 tập tin **myfile.txt** và nhập vào nội dung bất kỳ rồi kéo thả tập tin đó vào thư mục drawable. Nếu như ứng dụng không có thư mục drawable thì tự tạo ra thư mục này, tên file cũng không để khoảng trắng và không bắt đầu bằng chữ số.



5.2/ Xử lý tập tin

Bộ nhớ trong (Internal Storage) – Đọc tập tin trong Drawable.



1. Giao diện: [Activity_main.xml](#)

2. Viết đoạn [code](#) này vào sự kiện onClick của nút “Read Data”

5.2/ Xử lý tập tin

- Bộ nhớ trong (Internal Storage) – Đọc/Ghi dữ liệu.
- Internal storage là việc lưu trữ dữ liệu cục bộ theo từng ứng dụng trên bộ nhớ thiết bị. Dữ liệu sẽ bị mất khi người dùng xóa ứng dụng.
- Xử lý ghi dữ liệu vào bộ nhớ trong (Lưu ý, FileName là tên file mà bạn muốn tạo, str là dữ liệu bạn cần ghi vào bộ nhớ. Dữ liệu này chúng ta có thể lấy từ giao diện của ứng dụng)

```
OutputStream os = openFileOutput("FileName", MODE_APPEND/MODE_PRIVATE);  
String str = "Dữ liệu cần ghi";  
os.write(str.getBytes());  
os.close();
```

5.2/ Xử lý tập tin

- Bộ nhớ trong (Internal Storage) – Đọc/Ghi dữ liệu.
- Xử lý đọc dữ liệu từ bộ nhớ trong (Lưu ý, FileName là tên tập tin mà chúng ta cần đọc dữ liệu)

```
FileInputStream fis = openFileInput("fileName");  
BufferedReader br = new BufferedReader(new InputStreamReader(fis));  
StringBuffer data = new StringBuffer();  
String line = "";  
while ((line = br.readLine()) != null) {  
    data.append(line).append("\n");  
}
```

5.2/ Xử lý tập tin

Bộ nhớ ngoài (External Storage) – Đọc/Ghi dữ liệu.

Xử lý ghi dữ liệu vào bộ nhớ ngoài

```
File sdcard = Environment.getExternalStorageDirectory();  
File file = new File(sdcard, "FileName");  
OutputStream os = new FileOutputStream(file);  
String str = "Dữ liệu muốn ghi";  
os.write(str.getBytes());  
os.close();
```

5.2/ Xử lý tập tin

Bộ nhớ ngoài (External Storage) – Đọc/Ghi dữ liệu.

Xử lý đọc dữ liệu từ bộ nhớ ngoài

```
File sdcard = Environment.getExternalStorageDirectory();
File file = new File(sdcard,"FileName");
BufferedReader br = new BufferedReader(new FileReader(file));
String line;
//Đọc dữ liệu từ file, nội dung sau khi đọc sẽ chứa trong biến content
StringBuilder content = new StringBuilder();
while ((line = br.readLine()) != null) {
    content.append(line);
    content.append('\n');
}
br.close();
```


5.3/ XML Parser trong Android

- **XML** viết tắt của cụm từ “**Extensible Markup Language**” được dịch là “ngôn ngữ đánh dấu mở rộng“. Đây là một dạng ngôn ngữ đánh dấu có chức năng mô tả và truyền dữ liệu khác nhau
- **Tập tin XML** được dùng để mô tả dữ liệu. Các thẻ tag trong file XML chưa được xác định trước. Do đó, người dùng phải tự định nghĩa trong quá trình tạo file XML.
- **Tác dụng chính của file XML** là đơn giản việc chia sẻ tài nguyên dữ liệu giữa các platform và hệ thống khác nhau, đặc biệt những hệ thống được kết nối mạng. Chính vì thế, XML có tác dụng rất to lớn trong việc trao đổi, chia sẻ giữa các hệ thống.

5.3/ XML Parser trong Android

▪ Cấu trúc tập tin XML

```
<?xml version="1.0" encoding="UTF-8"?>
<công_thức_nấu_ăn tên="bánh mì" thời_gian_chuẩn_bị="5 phút" thời_gian_nấu="3 tiếng">
  <title>Bánh mì cơ bản</title>
  <nguyên_liệu lượng="3" đơn_vị="ca">Bột mì</nguyên_liệu>
  <nguyên_liệu lượng="7" đơn_vị="gram">Men</nguyên_liệu>
  <nguyên_liệu lượng="1.5" đơn_vị="ca" trạng_thái="ấm">Nước</nguyên_liệu>
  <nguyên_liệu lượng="1" đơn_vị="thìa cà phê">Muối</nguyên_liệu>
  <chỉ_dẫn>
    <bước>Trộn tất cả các nguyên liệu với nhau và nhào kĩ</bước>
    <bước>Phủ một mảnh vải, ủ một tiếng đồng hồ trong phòng ấm.</bước>
    <bước>Nhào lại, đổ vào khuôn, cho vào lò nướng.</bước>
  </chỉ_dẫn>
</công_thức_nấu_ăn>
```

Các website thường xuyên update nội dung như web tin tức , blog thông thường sẽ trả ra các file XML để cho các chương trình bên ngoài có thể đọc được những nội dung đó. 

5.3/ XML Parser trong Android

- Để đọc file có nội dung dạng chuẩn XML ta có 3 các Parser

	DOM	SAX	XMLPullParser
Cấu Trúc parser	Cấu trúc cây	Dựa vào kiểu sự kiện (không điều khiển được)	Dựa vào kiểu sự kiện (điều khiển được)
Truy xuất phần tử	Truy xuất bất kỳ	Truy xuất tuần tự	Truy xuất tuần tự
Lấy thông tin phần tử	Truy xuất cây DOM	Qua hàm callback	Truy xuất qua vòng lặp
Ghi vào tập tin	Có thể	Không thể	Có thể
Chi phí bộ nhớ	Nhiều	Ít nhất	Ít
Sử dụng	Dễ	Khó	Dễ

5.3/ XML Parser trong Android

XMLPullParser

- Là 1 trong số 3 cách Parser dữ liệu trên thì Android khuyến cáo chúng ta sử dụng XMLPullParser, bởi vì nó dễ sử dụng và tốn ít bộ nhớ.
- Cấu trúc của một file XML sẽ bao gồm những cặp thẻ và trong cặp thẻ đó có thể có các thuộc tính hoặc có thể có các giá trị nằm trong cặp thẻ

```
<?xml version="1.0"?>  
<?xml-stylesheet type="text/css" href="book.css"?>  
<list-book>  
  <book id="1">  
    <title>JP1</title>  
    <price>20</price>  
  </book>  
  <book id="2">  
    <title>JP2</title>  
    <price>25</price>  
  </book>  
</list-book>
```

Prolog

Events: gồm Document Start, document end, Tag start, Tag end, ...

Text

Attribute

5.3/ XML Parser trong Android

▪ XML - Parsing

Để có thể dùng `XmlPullParser` trước tiên chúng ta cần phải tạo ra một `XmlPullParser`, nhưng để tạo được `XmlPullParser` chúng ta cần phải tạo ra `XmlPullParserFactory` và gọi `newPullParser()` để tạo ra một `XmlPullParser`.

```
private XmlPullParserFactory xmlFactoryObject = XmlPullParserFactory.newInstance();  
private XmlPullParser myparser = xmlFactoryObject.newPullParser();
```

5.3/ XML Parser trong Android

- **XML - Parsing**

- Sau đó chúng ta cần chỉ định file nội dung cho XmlPullParser. Nội dung là một file XML

myParser.setInput(stream, null);

- Truy nội dung file XML:

```
int event = myParser.getEventType();
while (event != XmlPullParser.END_DOCUMENT) {
    String name = XmlPullParser.getName();
    Switch (event) {
        case XmlPullParser.START_TAG:
            ....
    }
```

5.3/ XML Parser trong Android

▪ XML - Parsing

<code>getEventType()</code>	Trả về kiểu của Event đang xảy ra. Ví dụ như: Document Start, Document End, ...
<code>getName()</code>	Trả về tên của Tag.
<code>getAttributeValue()</code>	Lấy các thuộc tính của Tag
<code>getAttributeCount()</code>	Trả về số thuộc tính có trong Tag.
<code>getAttributeName(int index)</code>	Trả về tên thuộc tính có index chỉ định
<code>getText()</code>	Trả về đoạn Text ở giữa thẻ

Câu hỏi Ôn tập

- 1) Đa ngôn ngữ trong ứng dụng Android là gì ?
- 2) Trình bày cách xử lý tập tin trong Android ?
- 3) XML Parser trong Android là gì?