

Hướng dẫn lập trình cơ bản với Android - Bài 2

Reflink: <http://diendan.vietandroid.com/cac-bai-hoc-co-kem-ma-nguon/409-huong-dan-lap-trinh-co-ban-voi-android-bai-2-a.html>

List tutorial

Bài 0 - Cài đặt và sử dụng Android với Eclipse

Bài 1 - Cơ bản Android

Bài 2 - Xây dựng giao diện đơn giản

Bài 3 - ViewGroup và Custom Adapter

Bài 4 - Intent và Broadcast Receiver

Bài 5 - Service

Bài 6 - SQLite

Bài 7 - Content Provider

Hi all,

Trong bài 1 mình đã giới thiệu sơ lược về các thành phần cơ bản của Android cũng như việc sử dụng XML để lập trình ứng dụng Android. Trong bài này mình sẽ giới thiệu thêm về Android Manifest và đi sâu hơn về vấn đề làm việc với View.

Android Manifest

Trong khung Package Explorer, ở phía dưới thư mục res, bạn sẽ thấy 1 file có tên là AndroidManifest.xml. Mỗi ứng dụng đều cần có AndroidManifest.xml để mô tả những thông tin quan trọng của nó cho hệ thống Android biết. Let's look closer:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="at.exam"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".Example"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Cụ thể những công việc mà AndroidManifest.xml thực hiện:

- Đặt tên cho Java package của ứng dụng.
- Mô tả các thành phần (component) của ứng dụng: activity, service, broadcast receiver hoặc content provider.
- Thông báo những permission mà ứng dụng cần có để truy nhập các protected API và tương tác với các ứng dụng khác.
- Thông báo những permission mà các ứng dụng khác cần có để tương tác với ứng dụng hiện thời.
- Thông báo level thấp nhất của Android API mà ứng dụng cần để chạy. (Android 1.0 là level 1, 1.1 là level 2, 1.5 level 3, 1.6 level 4 và 2.0 là level 5).

...

Hãy xem thử file AndroidManifest.xml của chương trình ToDo mình đang xây dựng:

www.Beenvn.com - Tủ Sách Online

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="android.at"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity
            android:name=".ToDo"
            android:screenOrientation="landscape"

            android:theme="@android:style/Theme.NoTitleBar.Fullscreen"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".WorkEnter">
        </activity>
        <receiver android:name=".AlarmReceiver">
        </receiver>
    </application>
    <uses-sdk android:minSdkVersion="3" />
    <uses-permission android:name="android.permission.VIBRATE" />
</manifest>
```

Main Activity của chương trình Too Do này là activity TooDo. Ngoài ra mình còn có 1 Activity khác có tên là WorkEnter để cho phép nhập vào thời gian và nội dung công việc. 1 Broadcast Receiver có tên là AlarmReceiver để nhận alarm gửi tới trong intent. Khi alarm được nhận sẽ có âm thanh và rung (vibration). Tất cả công việc sẽ được viết trong code, nhưng bắt buộc bạn phải khai báo các thành phần có trong ứng dụng vào AndroidManifest nếu muốn chương trình hoạt động. Tương tự, set permission để truy nhập camera, internet, đọc contact... cũng đều phải khai báo trong AM. Từ khóa **screenOrientation** cho phép thiết lập giao diện khi vào ứng dụng theo chiều dọc (portrait - mặc định) hay ngang (landscape), **theme** cho phép sử dụng style có sẵn của android là full-screen (không có thanh status bar nữa). Intent filter là bộ lọc dùng để giới hạn các intent được sử dụng trong activity hay receiver...

Mã:

```
<intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="http" android:host="www.google.com"
        android:path="/m/products/scan" />
</intent-filter>
```

Bộ lọc trên chỉ cho phép intent mở internet với đường dẫn định nghĩa sẵn (<http://www.google.com/m/products/scan>)

Ok, hi vọng mọi người đã nắm được chức năng cơ bản cũng như cách sử dụng Android

Manifest 

Working with View

Trong bài 1 mình đã giới thiệu qua cách sử dụng Edit Text và Text View. Thực chất các View còn lại cũng có cách sử dụng tương tự, bạn sẽ kết hợp nhiều View khác nhau để cho ra giao diện mình mong muốn. Ở đây mình sẽ đề cập nhiều tới List View (theo ý kiến mình là View khó sử

www.Beenvn.com - Tủ Sách Online

dụng nhất).

Yêu cầu: Xây dựng một chương trình cho phép nhập nội dung công việc và thời gian rồi list ra

B1: Vẫn bắt đầu bằng cách khởi tạo một Project mới: File -> New -> Android Project.

Project name: Example 2

Build Target: Chọn Android 1.5

Application name: Example 2

Package name: at.exam

Create Activity: Example

=> Kích nút Finish.

B2: Đi tới res/main.xml để xây dựng giao diện cho chương trình:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <EditText
        android:id="@+id/work_enter"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="@string/work_hint"
        android:lines="1"
        android:textSize="24px"
    />
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        >
        <TextView
            android:layout_width="50px"
            android:layout_height="wrap_content"
            android:text="@string/hour_edit"
            android:typeface="normal"
            android:textSize="15px"
            android:textStyle="bold"
            android:padding="5px"
        />
        <EditText
            android:id="@+id/hour_edit"
            android:layout_width="45px"
            android:layout_height="wrap_content"
            android:hint="12"
            android:textColorHint="@color/hint_color"
            android:textSize="20px"
            android:gravity="center"
            android:padding="5px"
            android:numeric="integer"
            android:maxLength="2"
        />
    <TextView
```

```

        android:layout_width="65px"
        android:layout_height="wrap_content"
        android:text="@string/minute_edit"
        android:typeface="normal"
        android:textSize="15px"
        android:textStyle="bold"
        android:padding="5px"
    />
    <EditText
        android:id="@+id/minute_edit"
        android:layout_width="45px"
        android:layout_height="wrap_content"
        android:hint="00"
        android:textColorHint="@color/hint_color"
        android:textSize="20px"
        android:gravity="center"
        android:padding="5px"
        android:numeric="integer"
        android:maxLength="2"
    />
</LinearLayout>
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/button_content"
/>
<ListView
    android:id="@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
/>
</LinearLayout>

```

Giao diện ta thiết kế ở đây có 1 Linear Layout làm thành phần chính, các thành phần con của nó gồm 1 Edit Text (dùng để nhập nội dung công việc), 1 Linear Layout (lại gồm các thành phần con để nhập giờ và phút thực hiện công việc), 1 Button (để thêm nội dung công việc vào List View) và 1 List View dùng để list các công việc bạn đã nhập.

Từ khóa **lines** được dùng để cố định số dòng và nên sử dụng với Edit Text thay vì dùng mỗi wrap_content vì nếu sd wrap_content thì Edit Text sẽ tự giãn ra nếu dòng nhập vào vượt giới hạn đường bao (làm hỏng giao diện bạn thiết kế).

Từ khóa **gravity** thông báo các thành phần con sẽ được sắp xếp ntn ở thành phần cha. Ở đây mình dùng "center" nghĩa là thành phần con nằm ở trung tâm. Hãy thử thêm vào 1 Edit Text: Mã:

```
android:gravity="center"
```

Bạn sẽ thấy dòng chữ nhập vào sẽ bắt đầu từ giữa của Edit Text chứ không bắt đầu từ bên trái như trước nữa.

Từ khóa **padding** dùng để cách 1 khoảng cách cho thành phần. Nếu không có padding thì 2 thành phần con thuộc cùng 1 LinearLayout sẽ được xếp sát nhau, nhưng nếu 1 thành phần con sử dụng padding thì sẽ tạo được khoảng cách với thành phần còn lại theo mong muốn. Ngoài ra còn có **paddingLeft**, **paddingRight**, **paddingTop**, **paddingBottom**.

Từ khóa **numeric** dùng để giới hạn dạng ký tự nhập vào. Ở đây mình muốn chỉ nhập vào chữ số nên dùng "integer"

Từ khóa **maxLength** dùng để giới hạn số ký tự nhập vào. Do Edit Text này dùng để nhập giờ nên

maxLength="2".

Ok, giờ đến 1 chút kiến thức về các đơn vị của dimension:

- px (pixel): điểm chấm trên màn hình.
 - in (inch)
 - mm (milimet)
 - pt (point) = 1/72 in
 - dp (density - independent pixel): cái này hơi khó giải thích. Nói chung dp được sử dụng cho nhiều độ phân giải, và với độ phân giải 160 px/inch thì 1 dp = 1 px.
 - sp: gần giống dp, nên sử dụng cho text size.
- Nói chung nên sử dụng dp và sp để định nghĩa size cho các thành phần, vì nó có tỉ lệ cố định với độ phân giải của màn hình. Còn nếu bạn chủ tâm xây dựng cho 1 độ phân giải nhất định thì dùng px cho chính xác và chắc chắn.

B3: Tới values/strings.xml chỉnh sửa như sau:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Example 2</string>
    <string name="work_hint">Enter the work here</string>
    <string name="hour_edit">Hour</string>
    <string name="minute_edit">Minute</string>
    <string name="button_content">Add work</string>
</resources>
```

B4: Tạo mới colors.xml trong values với nội dung:

Mã:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="hint_color">#cccccc</color>
</resources>
```

OK, vậy là đã hoàn thiện phần giao diện. Các bạn có thể cho chạy thử ngay để kiểm tra xem giao diện đã như ý muốn chưa chứ không cần đợi hoàn thành cả code (Run as -> Android Application).

B5: Time to coding. Tới thư mục src/Example.java và thay đổi nội dung file như sau:

Mã:

```
package at.exam;

import java.util.ArrayList;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

public class Example extends Activity {
    /** Called when the activity is first created. */
    @Override
```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    //Tạo mảng để chứa String nội dung công việc và giờ
    final ArrayList<String> arrayWork = new ArrayList<String>();
    //Adapter dùng để kết nối mảng với List View
    final ArrayAdapter<String> arrayAdapter = new
ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, arrayWork);

    //Các EditText để vào nội dung công việc được lấy về từ XML
    final EditText workEnter = (EditText)
findViewById(R.id.work_enter);
    final EditText hourEdit = (EditText)
findViewById(R.id.hour_edit);
    final EditText minuteEdit = (EditText)
findViewById(R.id.minute_edit);

    //Button khi nhấn sẽ thêm công việc vào ListView
    final Button button = (Button) findViewById(R.id.button);

    //ListView chứa danh sách công việc
    final ListView list = (ListView) findViewById(R.id.list);
    //Cần set Adapter cho list để biết sẽ lấy nội dung từ mảng
arrayWork
    list.setAdapter(arrayAdapter);

    //Định nghĩa Listener xử lý sự kiện nhấn vào button
    OnClickListener add = new OnClickListener() {
        @Override
        public void onClick(View v) {
            //Nếu 1 trong 3 Edit Text không có nội dung thì hiện
lên thông báo
            if (workEnter.getText().toString().equals("") ||
                hourEdit.getText().toString().equals("") ||
                minuteEdit.getText().toString().equals("")) {
                AlertDialog.Builder builder = new
AlertDialog.Builder(Example.this);
                builder.setTitle("Info missing");
                builder.setMessage("Please enter all information of
the work");
                builder.setPositiveButton("Continue", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int
which) {
                        // TODO Auto-generated method stub
                    }
                });
                builder.show();
            }
            //Lấy nội dung công việc và thời gian ra từ Edit Text
và đưa vào list
            else {
                String str = workEnter.getText().toString() + " - "

```

```

        + hourEdit.getText().toString() +
        + minuteEdit.getText().toString();
        arrayWork.add(0, str);
        arrayAdapter.notifyDataSetChanged();
        workEnter.setText("");
        hourEdit.setText("");
        minuteEdit.setText("");
    }
}

//set Listener cho button
button.setOnClickListener(add);
}
}

```

Mình đã chú thích đầy đủ và đoạn code cũng khá dễ hiểu. Tuy nhiên cần lưu ý 2 vấn đề ở đây.

- Khởi tạo đối tượng ArrayAdapter: Các bạn thấy đối số truyền vào là **(this, android.R.layout.simple_list_item_1, arrayWork)**. This là đối số của lớp Context (ở đây chính là activity Example). Bạn sẽ gặp Context trong rất nhiều khởi tạo các lớp và nên hiểu Context có ý nghĩa gì. Mình xin đưa ra giải thích của anh Giáp (thank mr giaplv):

Quote:

Context thuộc **android.content** (*android.content.Context*).

Là một Interface (lớp giao tiếp) chứa hầu hết thông tin về môi trường ứng dụng của android, có nghĩa là mọi thao tác, tương tác với hệ điều hành đều phải qua lớp này.

Nó là một lớp abstract (trừu tượng) cung cấp cho những lớp khác các phương thức để tương tác với hệ thống Android.

Nó cho phép truy cập tới các nguồn tài nguyên (resources) đã được định nghĩa và các lớp khác. Ví dụ như nó có thể khởi tạo và chạy các activities, các broadcast và các intents,... Chúng ta coi như Context là một lớp ở mức ứng dụng (Application level- liên quan tới hệ thống).

Tóm lại context giúp chúng ta dễ dàng truy cập và tương tác tới các tài nguyên của hệ thống, các thông tin, các dịch vụ (services), các thông số cấu hình, database, wallpaper, danh bạ, cuộc gọi, kết nối, chế độ rung (vibrator),...

***sở dĩ hầu hết các lớp có liên quan tới UI (layout, button, textview, imageview, listview,...) đều phải super tới Context vì bản thân nó đảm nhiệm việc truy cập resource (R.id, R.layout,...). Nếu chúng ta không tham chiếu tới Context class thì đương nhiên không thể dùng tới các resources mà chúng ta đã tạo ra.

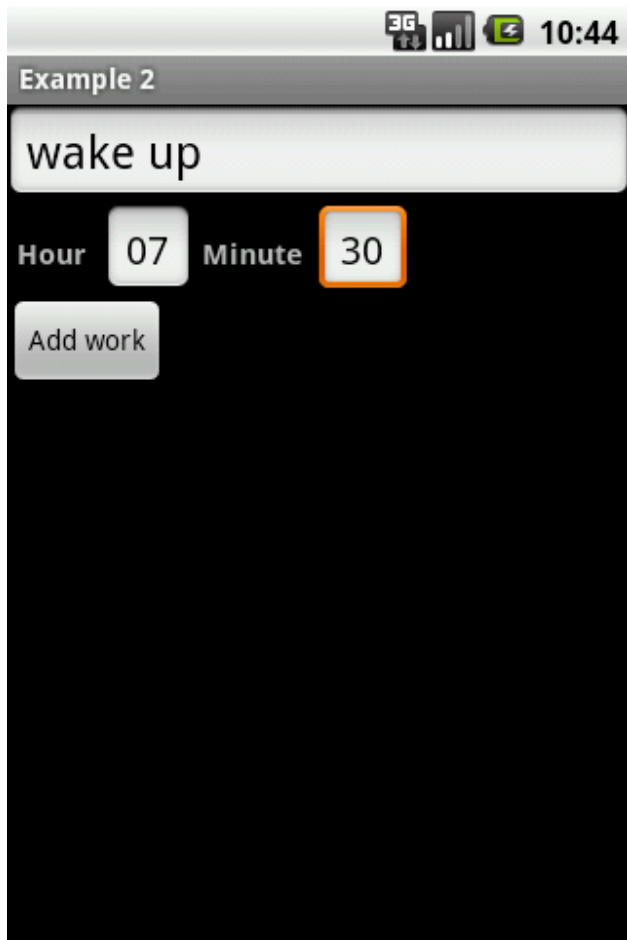
Tiếp theo là android.R.layout.simple_list_item_1, đối này định nghĩa cách thể hiện item (ở đây là String) trong List View. Các bạn hãy ghi nhớ android.R.* là các tài nguyên (resource) có sẵn của Android cho phép bạn truy cập và sử dụng. Sau này khi hướng dẫn tạo custom View cho List View mình sẽ đề cập lại vấn đề này.

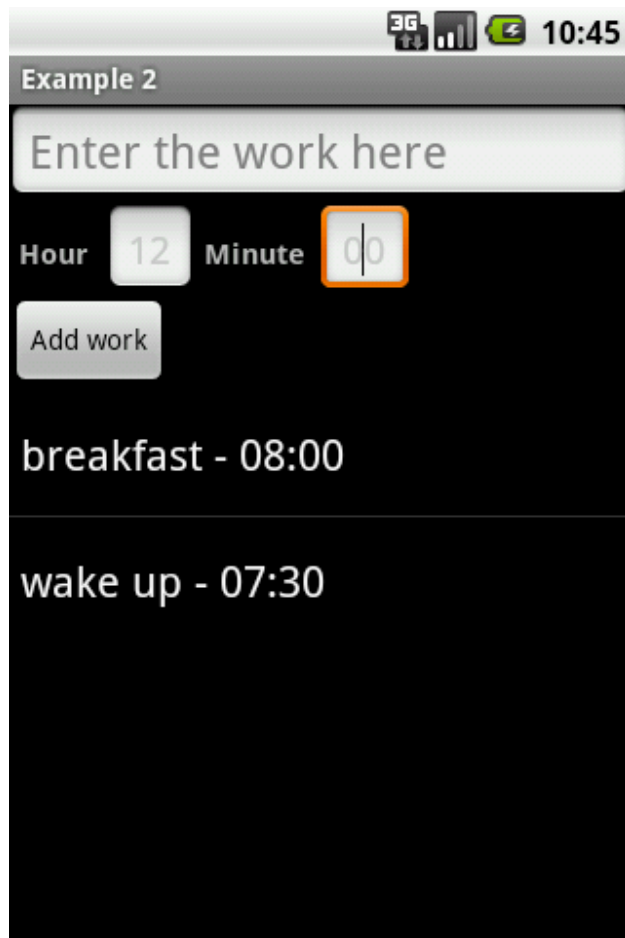
Cuối cùng arrayWork chính là mảng cần được bind của adapter.

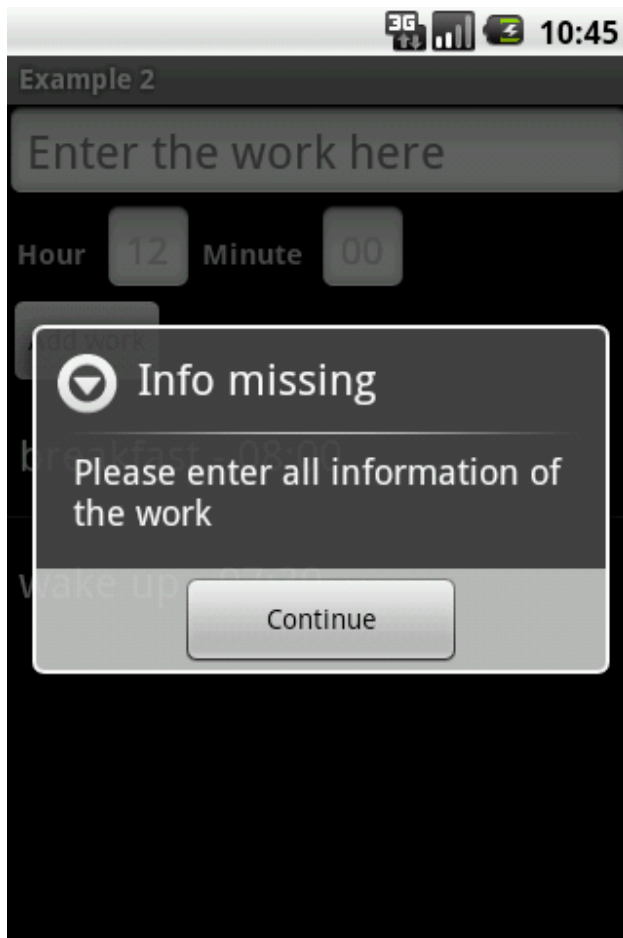
- AlertDialog là lớp cho phép đưa ra 1 hộp thoại, thường dùng để đưa ra thông tin hoặc cảnh báo đơn giản. Trong code mình tạo 1 builder, tạo tiêu đề (title) cho nó, đưa ra thông báo (message) và cuối cùng là tạo 1 positive button (nhưng không định nghĩa xử lý khi nhấn nút này, vì vậy nếu bạn nhấn nút thì dialog sẽ chỉ đơn giản thực hiện việc đóng lại).

B6: Tiến hành chạy thử chương trình. Run as -> Android Application. Enjoy yourself 🍷

www.Beenvn.com - Tủ Sách Online







Dự định bài tiếp theo sẽ hướng dẫn mọi người cách tạo custom view trong list view và option menu. 😊