Android 高手进阶教程

目录

Android 常用名令集锦

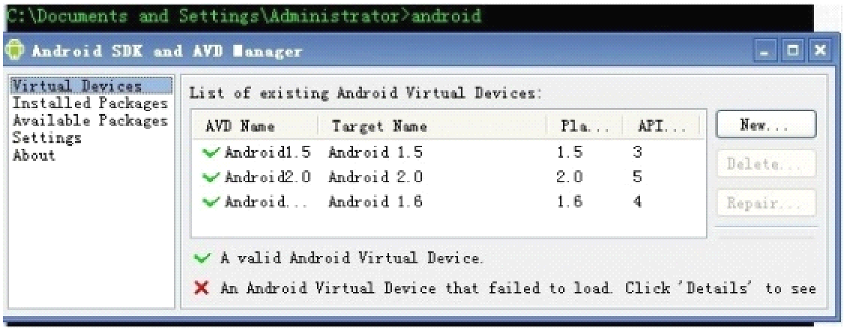
大家好，今天我们要讲的是 android 开发中，比较常用的名令集锦， 在我们开发中难免用到 Android 命令，有些确实命令确实很有用处。

特别对于一些初学者来说，命令根本没有想过用也不会用，比如他们想安装一个.apk 文件到模拟器上面，但是他们不会启动模拟器，他们只会先启动 Eclipse，然后在启动模拟器，这样不但浪费时间，而且 Eclipse 又占用你的内存。这也是我为什么总结这篇文章的原因了，希望对大家有所帮助。

如果想让系统认识你输入的命令(如:输入 android 命令)有两种方法：1.设置环境变量.(和设置 java 路径一样，具体网上查哦!);2.直接进 入你 SDK 里 tools 目录(cd:sdkpath/tools)这样也 OK.不过在此建议使用第一种!下面就是自己的小小总结:

1.android:

对你只要输入 android 就会出来,SDK and AVD manager 我们可以更新 SDK,增删修改 AVD.效果如下图:



```
C:\Documents and Settings\Administrator>android
```

List of existing Android Virtual Devices:

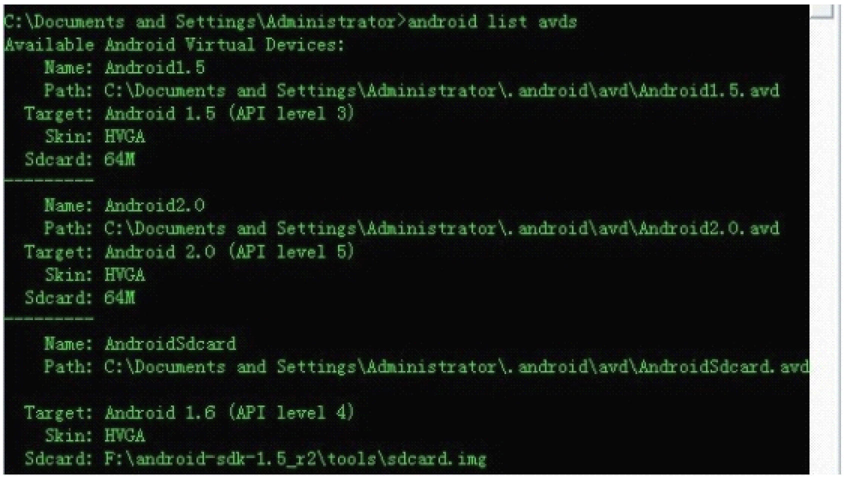
AVD Name	Target Name	Pla...	API...
✓ Android1.5	Android 1.5	1.5	3
✓ Android2.0	Android 2.0	2.0	5
✓ Android...	Android 1.6	1.6	4

✓ A valid Android Virtual Device.

✗ An Android Virtual Device that failed to load. Click 'Details' to see

2.android list avds:

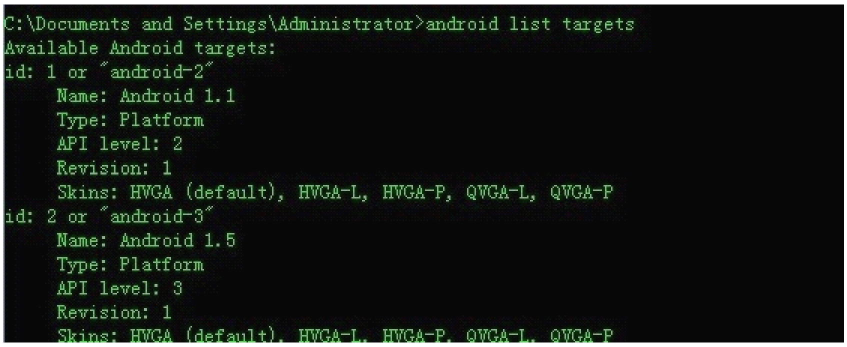
这条命令将会列出所有我们创建的 android 模拟器.效果如下图:



```
C:\Documents and Settings\Administrator>android list avds
Available Android Virtual Devices:
    Name: Android1.5
    Path: C:\Documents and Settings\Administrator\.android\avd\Android1.5.avd
  Target: Android 1.5 (API level 3)
    Skin: HVGA
  Sdcard: 64M
---------
    Name: Android2.0
    Path: C:\Documents and Settings\Administrator\.android\avd\Android2.0.avd
  Target: Android 2.0 (API level 5)
    Skin: HVGA
  Sdcard: 64M
---------
    Name: AndroidSdcard
    Path: C:\Documents and Settings\Administrator\.android\avd\AndroidSdcard.avd

  Target: Android 1.6 (API level 4)
    Skin: HVGA
  Sdcard: F:\android-sdk-1.5_r2\tools\sdcard.img
```

3.android list targets:

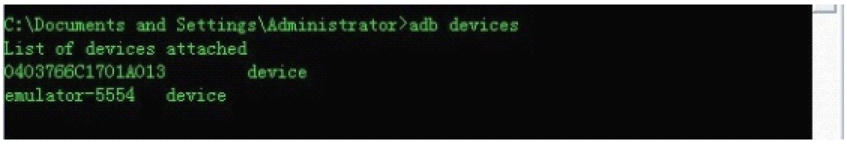
这条命令是列出我们所有的 SDK 可用版本,效果如下图:



```
C:\Documents and Settings\Administrator>android list targets
Available Android targets:
id: 1 or "android-2"
     Name: Android 1.1
     Type: Platform
     API level: 2
     Revision: 1
     Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
id: 2 or "android-3"
     Name: Android 1.5
     Type: Platform
     API level: 3
     Revision: 1
     Skins: HVGA (default), HVGA-L, HVGA-P, QVGA-L, QVGA-P
```

4.adb devices:

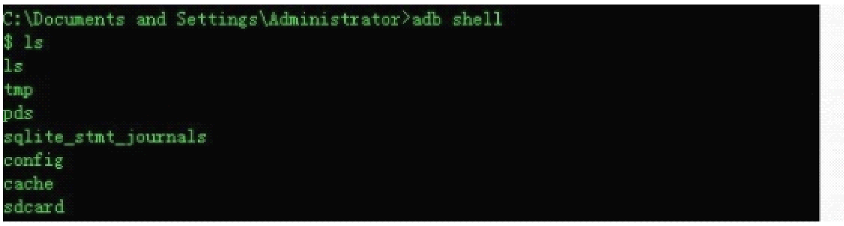
这条命令是列出所有装载的设置，效果如下图(一个是模拟器，一个是真机):



```
C:\Documents and Settings\Administrator>adb devices
List of devices attached
0403766C1701A013        device
emulator-5554   device
```

5.adb shell:

这条命令是进入设备根目录/,取得对设备的控制权，如输入 ls 命令等。



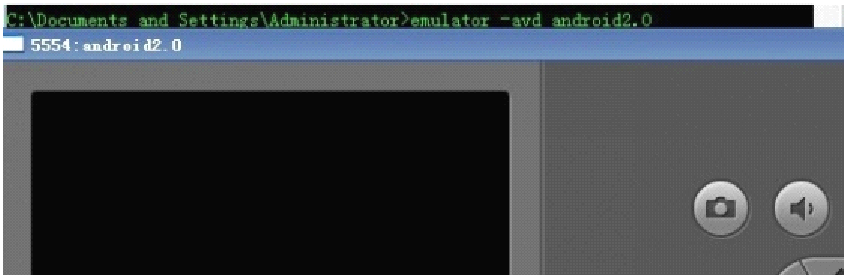
```
C:\Documents and Settings\Administrator>adb shell
$ ls
ls
tmp
pds
sqlite_stmt_journals
config
cache
sdcard
```

6.adb install XXX.apk.

这条命令是安装 apk 文件，如果你有多个设备(而你想把 apk 安装到 emulator-5554 这个模拟器上)则要输入:adb install -s emulator-5554 D:/XXX.apk.

7.emulator -avd avdname

这条命令将启动一个模拟器，初学者学会这条，就不用下次启动模拟器的时候还要打开 Eclipse 了。用法如下:



```
C:\Documents and Settings\Administrator>emulator -avd android2.0
```

8.mksdcard 256M d:\sdcard.img

对于模拟器当然要创建 sdcard 了，这条命令是将在 D 盘下生成 256M 的 sdcard.

9.adb pull <remote> <local>/adb push <local> <remote>.

我们创建完 sdcard 以后，要向里面放东西，或者把里面的东西弄出来，就用到以上命令了。以 adb push(将 sdcard 外的东西存放进去.)为例，用法如下:adb push foo.txt /sdcard/foo.txt

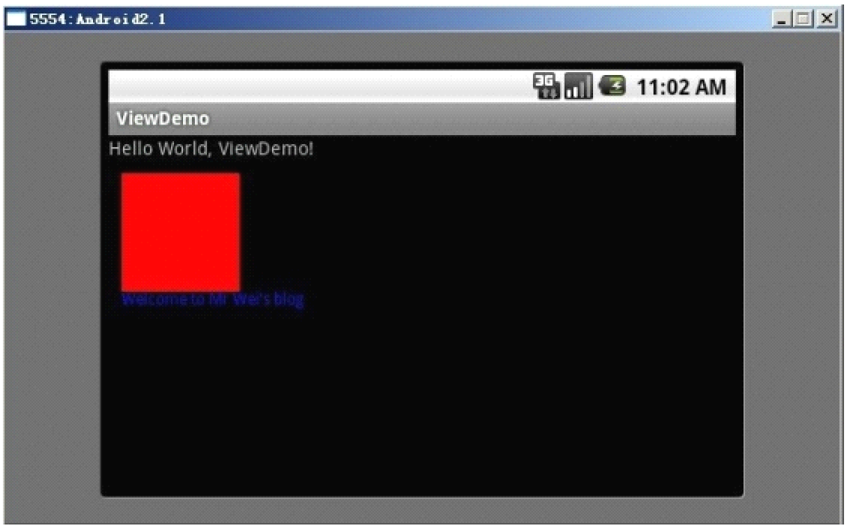
10.android create avd --name Android2.0 --target 5

这是创建 avd 的命令。--name 后面就是 avd 的名称，target 是 SDK 的版本。这条语句是创建名称为 Android2.0 并且版本为 5 的 avd.

创建完可以用 android list avds。查看是否已经创建成功。

11.ctrl + F11 键的使用

当我们启动模拟器的时候,通常是 port 模式，如果我们想在 land 下看效果,这个组合按钮就用到啦.看一下效果图：



Hello World, ViewDemo!

Welcome to Mr Wei's blog

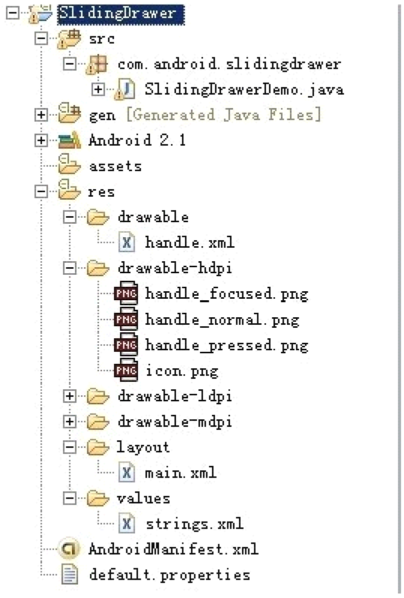
Android Launcher 抽屉类 SlidingDrawer 的使用

最近在研究 Lanucher ,看了源码，发现了 SlidingDrawer 这个类，也就是所谓的"抽屉"类。它的用法很简单，要包括 handle ,和 content .

handle 就是当你点击它的时候，content 要么抽抽屉要么关抽屉。别的不多说了，具体步骤如下.

1.新建 Android 工程,命名为 SlidingDrawer .

2.准备素材,在这里我的图标是用 Launcher2 里面的图标,放在 drawable-hdpi 文件夹目录结构如下:



```
SlidingDrawer
  src
    com.android.slidingdrawer
      SlidingDrawerDemo.java
  gen [Generated Java Files]
  Android 2.1
  assets
  res
    drawable
      handle.xml
    drawable-hdpi
      handle_focused.png
      handle_normal.png
      handle_pressed.png
      icon.png
    drawable-ldpi
    drawable-mdpi
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  default.properties
```

3.设置 main.xml 布局:代码如下:

view plaincopy to clipboardprint?

```
<?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"
      android:background="#808080"
      >
<SlidingDrawer
      android:id="@+id/slidingdrawer"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:orientation="vertical"
      android:handle="@+id/handle"
      android:content="@+id/content">
      <Button
                    android:id="@+id/handle"
                    android:layout_width="88dip"
                    android:layout_height="44dip"
                    android:background="@drawable/handle"
              />
      <LinearLayout
              android:id="@+id/content"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:background="#00ff00">
              <Button
                    android:id="@+id/button"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:text="Button"
              />
              <EditText
```

```
                    android:id="@+id/editText"
                    android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
                />
      </LinearLayout>
</SlidingDrawer>
</LinearLayout>
<?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"
      android:background="#808080"
      >
<SlidingDrawer
      android:id="@+id/slidingdrawer"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      android:orientation="vertical"
      android:handle="@+id/handle"
      android:content="@+id/content">
      <Button
                    android:id="@+id/handle"
                    android:layout_width="88dip"
                    android:layout_height="44dip"
                    android:background="@drawable/handle"
                />
      <LinearLayout
              android:id="@+id/content"
```

```
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="#00ff00">
        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button"
        />
        <EditText
            android:id="@+id/editText"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
        />
    </LinearLayout>
</SlidingDrawer>
</LinearLayout>
```

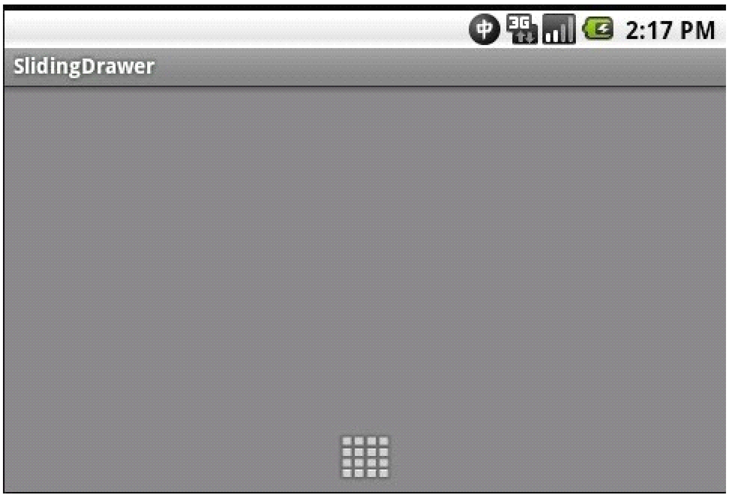
4.设置 handle 图标的样式，在 drawable 里添加 handle.xml ,代码如下:

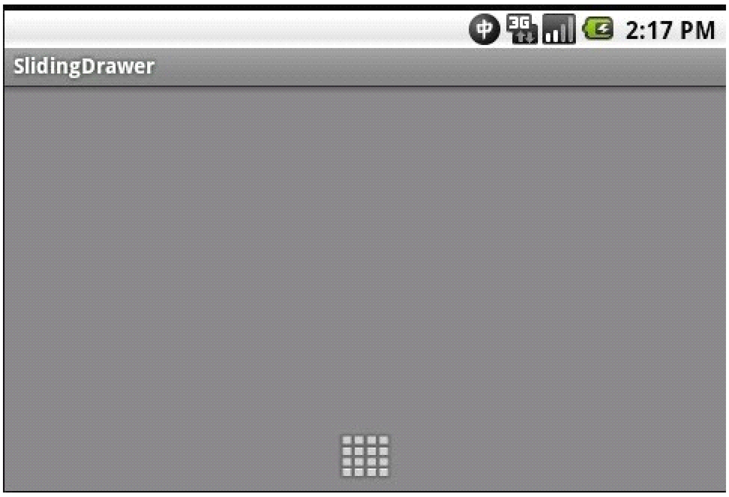
view plaincopy to clipboardprint?

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_window_focused="false" android:state_enabled="true"
android:drawable="@drawable/handle_normal" />
    <item android:state_pressed="true"
android:drawable="@drawable/handle_pressed" />
    <item android:state_focused="true" android:state_enabled="true"
android:drawable="@drawable/handle_focused" />
    <item android:state_enabled="true"
android:drawable="@drawable/handle_normal" />
```

```
      <item android:state_focused="true"
android:drawable="@drawable/handle_focused" />
</selector>
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
      <item android:state_window_focused="false" android:state_enabled="true"
android:drawable="@drawable/handle_normal" />
      <item android:state_pressed="true"
android:drawable="@drawable/handle_pressed" />
      <item android:state_focused="true" android:state_enabled="true"
android:drawable="@drawable/handle_focused" />
      <item android:state_enabled="true"
android:drawable="@drawable/handle_normal" />
      <item android:state_focused="true"
android:drawable="@drawable/handle_focused" />
</selector>
```

5.运行之。将会得到如下效果:





的比较简单呵呵，如果想深入了解，大家看 Launcher 源码吧！

Android 中自定义 View 的应用

大家好我们今天的教程是在 Android 教程中自定义 View 的学习，对于初学着来说,他们习惯了 Android 传统的页面布局方式，如下代码:

view plaincopy to clipboardprint?

```
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
</LinearLayout>
```

```
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
```

```
</LinearLayout>
```

当然上面的布局方式可以帮助我们完成简单应用的开发了，但是如果你想写一个复杂的应用，这样就有点牵强了，大家不信可以下源码都研究看看，高手写的布局方式,如上面的布局高手通常是这样写的:

view plaincopy to clipboardprint?

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

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

view plaincopy to clipboardprint?

其中 A extends LinerLayout, B extends TextView.

其中 A extends LinerLayout, B extends TextView.

为了帮助大家更容易理解，我写了一个简单的Demo ,具体步骤如下：

首先新建一个Android 工程 命名为ViewDemo .

然后自定义一个View 类，命名为MyView(extends View) .代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.Paint.Style;
import android.util.AttributeSet;
import android.view.View;
```

```
public class MyView extends View {
      private Paint mPaint;
      private Context mContext;
      private static final String mString = "Welcome to Mr Wei’s blog";

      public MyView(Context context) {
              super(context);

      }
      public MyView(Context context,AttributeSet attr)
      {
              super(context,attr);

      }
      @Override
      protected void onDraw(Canvas canvas) {
              // TODO Auto-generated method stub
              super.onDraw(canvas);

              mPaint = new Paint();

              //设置画笔颜色
              mPaint.setColor(Color.RED);
              //设置填充
              mPaint.setStyle(Style.FILL);

              //画一个矩形,前俩个是矩形左上角坐标，后面俩个是右下角坐标
              canvas.drawRect(new Rect(10, 10, 100, 100), mPaint);
              mPaint.setColor(Color.BLUE);
              //绘制文字
              canvas.drawText(mString, 10, 110, mPaint);
      }
}
package com.android.tutor;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.Paint.Style;
import android.util.AttributeSet;
import android.view.View;
public class MyView extends View {
private Paint mPaint;
private Context mContext;
private static final String mString = "Welcome to Mr Wei's blog";

public MyView(Context context) {
    super(context);

}
public MyView(Context context,AttributeSet attr)
{
    super(context,attr);

}
@Override
```

```
protected void onDraw(Canvas canvas) {
    // TODO Auto-generated method stub
    super.onDraw(canvas);

    mPaint = new Paint();

    //设置画笔颜色
    mPaint.setColor(Color.RED);
    //设置填充
    mPaint.setStyle(Style.FILL);

    //画一个矩形,前俩个是矩形左上角坐标，后面俩个是右下角坐标
    canvas.drawRect(new Rect(10, 10, 100, 100), mPaint);

    mPaint.setColor(Color.BLUE);
    //绘制文字
    canvas.drawText(mString, 10, 110, mPaint);
}
}
```

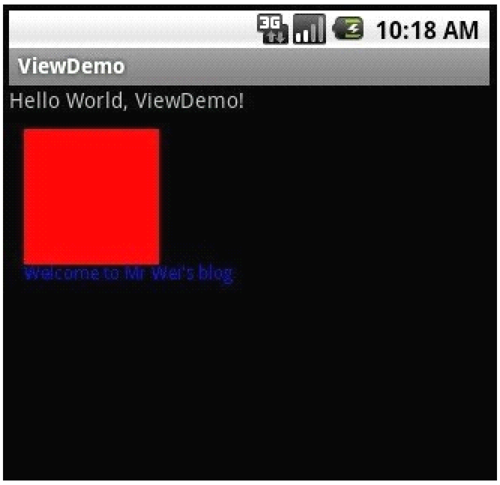
然后将我们自定义的 View 加入到 main.xml 布局文件中,代码如下:

view plaincopy to clipboardprint?

```
<?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"
      >
<TextView
```

```
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<com.android.tutor.MyView
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
/>
</LinearLayout>
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<com.android.tutor.MyView
android:layout_width="fill_parent"
      android:layout_height="fill_parent"
/>
</LinearLayout>
```

最后执行之，效果如下图:



Hello World, ViewDemo!

Welcome to Mr Wei's blog

OK,大功告成，今天就写到这里

Android 中自定义属性(attr.xml,TypedArray)的使用

今天我们的教程是根据前面一节扩展进行的，如果你没有看,请点击 Android高手进阶教程(三) 查看第三课,这样跟容易方便你的理解!

在xml 文件里定义控件的属性，我们已经习惯了android:attrs="" ,那么我们能不能定义自己的属性能，比如:test:attrs="" 呢？答案是肯定的.

好了我就不卖关子了，直接进入主题。大致以下步骤:

一、 在res/values 文件下定义一个attrs.xml 文件.代码如下:

view plaincopy to clipboardprint?

一、在res/values文件下定义一个attrs.xml文件.代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="MyView">
        <attr name="textColor" format="color" />
        <attr name="textSize" format="dimension" />
    </declare-styleable>
</resources>
```

一、在res/values文件下定义一个attrs.xml文件.代码如下:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="MyView">
        <attr name="textColor" format="color" />
        <attr name="textSize" format="dimension" />
    </declare-styleable>
</resources>
```

二、 我们在MyView.java 代码修改如下，其中下面的构造方法是重点，我们获取定义的属性我们R.sytleable.MyView_textColor, 获取方法中后面通常设定默认值(float textSize = a.getDimension(R.styleable.MyView_textSize, 36);)， 防止我们在xml 文件中没

有定义.从而使用默认值!

获取，MyView 就是定义在<declare-styleable name="MyView
"></declare-styleable> 里的名字,获取里面属性用 名字_ 属性 连接起来就可以.TypedArray 通常最后调用 .recycle() 方法，为了保持以后使用该属性一致性!

view plaincopy to clipboardprint?

```
public MyView(Context context,AttributeSet attrs)
      {
              super(context,attrs);
              mPaint = new Paint();

              TypedArray a = context.obtainStyledAttributes(attrs,
                          R.styleable.MyView);

              int textColor = a.getColor(R.styleable.MyView_textColor,
                          0XFFFFFFFF);
              float textSize = a.getDimension(R.styleable.MyView_textSize,
36);

              mPaint.setTextSize(textSize);
              mPaint.setColor(textColor);

              a.recycle();
      }
public MyView(Context context,AttributeSet attrs)
{
    super(context,attrs);
    mPaint = new Paint();

    TypedArray a = context.obtainStyledAttributes(attrs,
      R.styleable.MyView);

    int textColor = a.getColor(R.styleable.MyView_textColor,
      0XFFFFFFFF);
    float textSize = a.getDimension(R.styleable.MyView_textSize, 36);

    mPaint.setTextSize(textSize);
    mPaint.setColor(textColor);

    a.recycle();
}
```

MyView.java 全部代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.Paint.Style;
import android.util.AttributeSet;
import android.view.View;
public class MyView extends View {
      private Paint mPaint;
      private Context mContext;
      private static final String mString = "Welcome to Mr Wei's blog";

      public MyView(Context context) {
              super(context);
              mPaint = new Paint();
      }
      public MyView(Context context,AttributeSet attrs)
      {
              super(context,attrs);
              mPaint = new Paint();

              TypedArray a = context.obtainStyledAttributes(attrs,
                          R.styleable.MyView);

              int textColor = a.getColor(R.styleable.MyView_textColor,
                          0XFFFFFFFF);
              float textSize = a.getDimension(R.styleable.MyView_textSize,
36);

              mPaint.setTextSize(textSize);
              mPaint.setColor(textColor);

              a.recycle();
      }
      @Override
      protected void onDraw(Canvas canvas) {
              // TODO Auto-generated method stub
              super.onDraw(canvas);
              //设置填充
              mPaint.setStyle(Style.FILL);

              //画一个矩形,前俩个是矩形左上角坐标，后面俩个是右下角坐标
              canvas.drawRect(new Rect(10, 10, 100, 100), mPaint);

              mPaint.setColor(Color.BLUE);
              //绘制文字
              canvas.drawText(mString, 10, 110, mPaint);
      }
}
package com.android.tutor;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Rect;
import android.graphics.Paint.Style;
import android.util.AttributeSet;
import android.view.View;
public class MyView extends View {
private Paint mPaint;
private Context mContext;
private static final String mString = "Welcome to Mr Wei's blog";

public MyView(Context context) {
    super(context);
    mPaint = new Paint();
}
public MyView(Context context,AttributeSet attrs)
{
    super(context,attrs);
    mPaint = new Paint();

    TypedArray a = context.obtainStyledAttributes(attrs,
      R.styleable.MyView);

    int textColor = a.getColor(R.styleable.MyView_textColor,
      0XFFFFFFFF);
    float textSize = a.getDimension(R.styleable.MyView_textSize, 36);

    mPaint.setTextSize(textSize);
    mPaint.setColor(textColor);

    a.recycle();
}
@Override
protected void onDraw(Canvas canvas) {
    // TODO Auto-generated method stub
    super.onDraw(canvas);
    //设置填充
    mPaint.setStyle(Style.FILL);

    //画一个矩形,前俩个是矩形左上角坐标，后面俩个是右下角坐标
    canvas.drawRect(new Rect(10, 10, 100, 100), mPaint);

    mPaint.setColor(Color.BLUE);
    //绘制文字
    canvas.drawText(mString, 10, 110, mPaint);
}
}
```

三、将我们自定义的 MyView 加入布局 main.xml 文件中，平且使用自定义属性,自定义属性必须加上:

xmlns:test ="http://schemas.android.com/apk/res/com.android.tutor "蓝色 是自定义属性的前缀，红色 是我们包名.

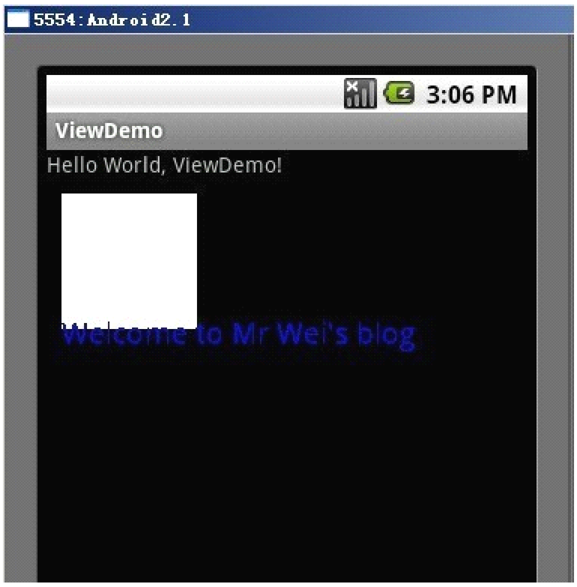
main.xml 全部代码如下:

view plaincopy to clipboardprint?

```
<?xml
version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:test="http://schemas.android.com/apk/res/com.android.tutor"
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<com.android.tutor.MyView
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      test:textSize="20px"
      test:textColor="#fff"
/>
</LinearLayout>
```

```
<?xml
version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:test="http://schemas.android.com/apk/res/com.android.tutor"
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<com.android.tutor.MyView
android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      test:textSize="20px"
      test:textColor="#fff"
/>
</LinearLayout>
```

四、运行之效果如下图:



Hello World, ViewDemo!

Welcome to Mr Wei's blog

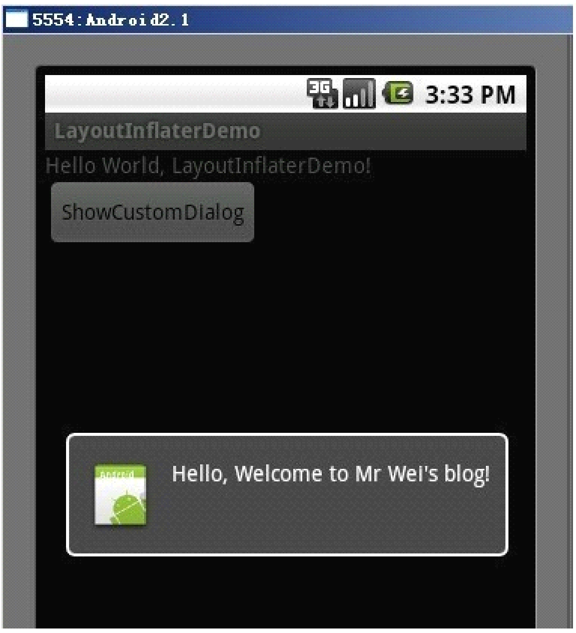
今天就到此结束

Android 中 LayoutInflater 的使用

大家好我们这一节讲的是 LayoutInflater 的使用,在实际开发种 LayoutInflater 这个类还是非常有用的,它的作用类似于 findViewById(),
不同点是LayoutInflater是用来找 layout 下 xml 布局文件,并且实例化!而 findViewById() 是找具体 xml 下的具体 widget 控件(如:Button,TextView 等)。

为了让大家容易理解我做了一个简单的 Demo,主布局 main.xml 里有一个 TextView 和一个 Button，当点击 Button，出现 Dialog,而这个 Dialog 的布局方式是我们在 layout 目录下定义的 custom_dialog.xml 文件(里面左右分布，左边 ImageView,右边 TextView)。

效果图如下:



Hello World, LayoutInflaterDemo!

Hello, Welcome to Mr Wei's blog!

下面我将详细的说明 Demo 的实现过程:

1、新建一个 Android 工程，我们命名为 LayoutInflaterDemo.

2、修改 main.xml 布局，里面主要在原来基础上增加了一个 Button.代码如下:

view plaincopy to clipboardprint?

```
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="ShowCustomDialog"
      />
</LinearLayout>
```

```
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      />
<Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="ShowCustomDialog"
/>
</LinearLayout>
```

3.定义对话框的布局方式，我们在 layout 目录下，新建一个名为 custom_dialog.xml 文件具体代码如下:

view plaincopy to clipboardprint?

```
<?xml version="1.0"
encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
                        android:orientation="horizontal"
                        android:layout_width="fill_parent"
                        android:layout_height="fill_parent"
                        android:padding="10dp"
                        >
      <ImageView android:id="@+id/image"
                         android:layout_width="wrap_content"
                         android:layout_height="fill_parent"
                         android:layout_marginRight="10dp"
                         />
      <TextView android:id="@+id/text"
                        android:layout_width="wrap_content"
                        android:layout_height="fill_parent"
                        android:textColor="#FFF"
                        />
</LinearLayout>
<?xml version="1.0"
encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
                        android:orientation="horizontal"
                        android:layout_width="fill_parent"
                        android:layout_height="fill_parent"
                        android:padding="10dp"
                        >
      <ImageView android:id="@+id/image"
                         android:layout_width="wrap_content"
                         android:layout_height="fill_parent"
                         android:layout_marginRight="10dp"
                         />
      <TextView android:id="@+id/text"
                        android:layout_width="wrap_content"
                        android:layout_height="fill_parent"
                        android:textColor="#FFF"
                        />
</LinearLayout>
```

4.修改主程序 LayouInflaterDemo.java 代码如下:

view plaincopy to clipboardprint?

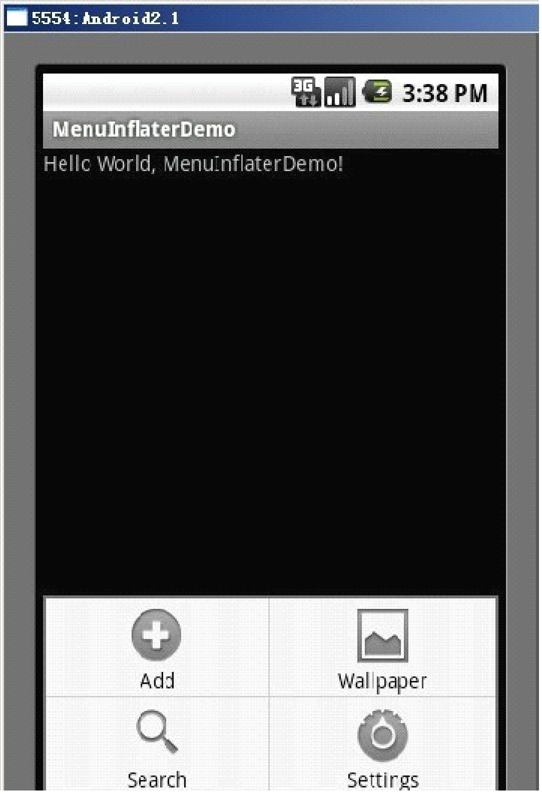
```
package com.android.tutor;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
public class LayoutInflaterDemo extends Activity implements
OnClickListener {

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

              button = (Button)findViewById(R.id.button);
              button.setOnClickListener(this);
      }
      @Override
      public void onClick(View v) {

              showCustomDialog();
      }

      public void showCustomDialog()
      {
              AlertDialog.Builder builder;
              AlertDialog alertDialog;
              Context mContext = LayoutInflaterDemo.this;

              //下面俩种方法都可以
              ////LayoutInflater inflater = getLayoutInflater();
              LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(LAYOUT_INFLATER_SERVICE);
              View layout = inflater.inflate(R.layout.custom_dialog,null);
              TextView text = (TextView) layout.findViewById(R.id.text);
              text.setText("Hello, Welcome to Mr Wei's blog!");
              ImageView image = (ImageView) layout.findViewById(R.id.image);

              image.setImageResource(R.drawable.icon);
              builder = new AlertDialog.Builder(mContext);
              builder.setView(layout);
              alertDialog = builder.create();
              alertDialog.show();
      }
}
package com.android.tutor;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
public class LayoutInflaterDemo extends Activity implements
OnClickListener {

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

              button = (Button)findViewById(R.id.button);
              button.setOnClickListener(this);
      }
@Override
public void onClick(View v) {

    showCustomDialog();
}

public void showCustomDialog()
{
    AlertDialog.Builder builder;
    AlertDialog alertDialog;
    Context mContext = LayoutInflaterDemo.this;
    //下面俩种方法都可以
    ////LayoutInflater inflater = getLayoutInflater();
    LayoutInflater inflater = (LayoutInflater)
mContext.getSystemService(LAYOUT_INFLATER_SERVICE);
    View layout = inflater.inflate(R.layout.custom_dialog,null);
    TextView text = (TextView) layout.findViewById(R.id.text);
    text.setText("Hello, Welcome to Mr Wei's blog!");
    ImageView image = (ImageView) layout.findViewById(R.id.image);
    image.setImageResource(R.drawable.icon);
    builder = new AlertDialog.Builder(mContext);
    builder.setView(layout);
    alertDialog = builder.create();
    alertDialog.show();
}
}
```

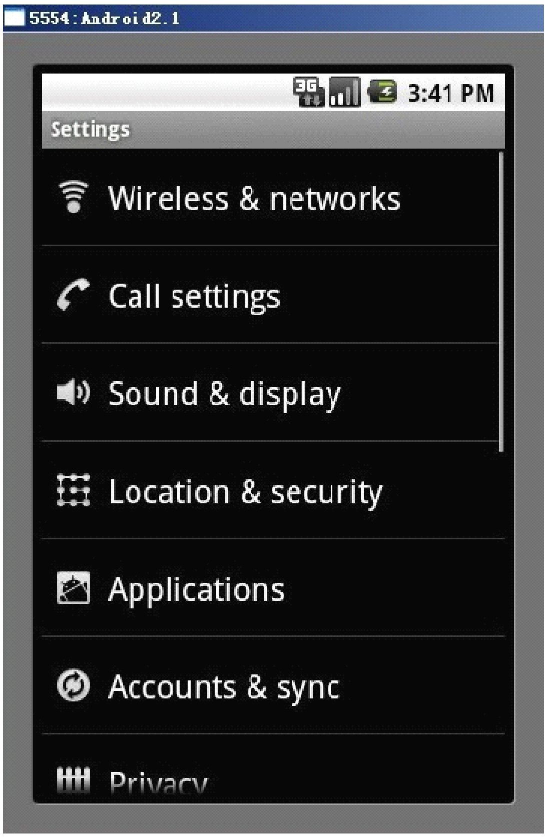
5、最后执行之，点击 Button，将得到上述效果。

Android 中 MenuInflater 的使用(布局定义菜单)

大家好，我们上一节讲的是 LayoutInflater 的 使用，而这一节我将讲一下 MenuInflater ,顾名思义，LayoutInflater 是用来解析定义在 layout 下的布局文件，那么 MenuInflater 是不是用来解析定义在 menu 目录下的菜单布局文件呢？ 恭喜你答对了!(*^__^*) 嘻嘻……

我们传统意义上 的定义菜单感觉比较繁琐，当我们使用 MenuInflater 来生成菜单，你会发现是多么的爽朗，呵呵，我今天的小 Demo,是定义四个菜单，并且实现了一个菜单事件。就是我们点击设置 (Setting)菜单，进入手机设置状态！下面看一下效果图:





下面是实现 Demo 的详细步骤:

一、建立一个 Android 工程我们命名为 MenuInflaterDemo .

二、在 res 目录下创建 menu 目录，并且创建 options_menu.xml (我们定义的菜单)文件，代码如下:

```
view plaincopy to clipboardprint?
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
      <item android:id="@+id/menu_add"
```

```
                android:title="Add"
                android:icon="@android:drawable/ic_menu_add"
                   />
      <item android:id="@+id/menu_wallaper"
                android:title="Wallpaper"
                android:icon="@android:drawable/ic_menu_gallery"
                   />
     <item android:id="@+id/menu_search"
                android:title="Search"
                android:icon="@android:drawable/ic_search_category_default"
                   />
     <item android:id="@+id/menu_setting"
                android:title="Settings"
                android:icon="@android:drawable/ic_menu_preferences"
                   />
</menu>
<?xml version="1.0" encoding="utf-8"?>
<menu
    xmlns:android="http://schemas.android.com/apk/res/android">
      <item android:id="@+id/menu_add"
                android:title="Add"
                android:icon="@android:drawable/ic_menu_add"
                   />
      <item android:id="@+id/menu_wallaper"
                android:title="Wallpaper"
                android:icon="@android:drawable/ic_menu_gallery"
                   />
     <item android:id="@+id/menu_search"
                android:title="Search"
                android:icon="@android:drawable/ic_search_category_default"
                   />
     <item android:id="@+id/menu_setting"
                android:title="Settings"
                android:icon="@android:drawable/ic_menu_preferences"
                   />
</menu>
```

三、主类 MenuInflaterDemo.java 的编码，这里写的代码很少哦，我这里只写了第四个菜单(Settings)的响应事件.全部代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
public class MenuInflaterDemo extends Activity {
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
      }

      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
              MenuInflater inflater = getMenuInflater();
              inflater.inflate(R.menu.options_menu, menu);
```

```
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case R.id.menu_add:

            break;
        case R.id.menu_wallaper:
            break;
        case R.id.menu_search:
            break;
        case R.id.menu_setting:
            showSettings();
            break;
        }
        return super.onOptionsItemSelected(item);
    }


    private void showSettings(){

                            final Intent settings = new
Intent(android.provider.Settings.ACTION_SETTINGS);
            settings.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
                            Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
```

```
              startActivity(settings);
     }
}
package com.android.tutor;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
public class MenuInflaterDemo extends Activity {
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
      }

      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
         MenuInflater inflater = getMenuInflater();
         inflater.inflate(R.menu.options_menu, menu);
         return true;
      }

      @Override
      public boolean onOptionsItemSelected(MenuItem item) {
         switch (item.getItemId()) {
    case R.id.menu_add:
     break;
    case R.id.menu_wallaper:
     break;
    case R.id.menu_search:
     break;
    case R.id.menu_setting:
     showSettings();
     break;
    }
        return super.onOptionsItemSelected(item);
      }


      private void showSettings(){

                                 final      Intent      settings      =      new
Intent(android.provider.Settings.ACTION_SETTINGS);
               settings.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
                             Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);

        startActivity(settings);
     }
}
```

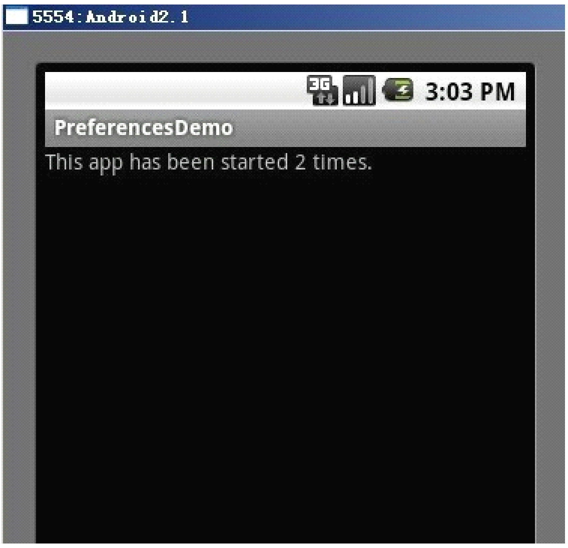
四、运行代码之，点击模拟器上的 menu 按钮将会出现上述效果图!

Android 中 Preferences 的使用!

大家好,我们这一节讲的是Android Preferences 的学习,Preferences 在Android 当中被用来记录应用，以及用户喜好等等，它可以用来保存

简单的数据类型，如 Int,Double,Boolean 等。Preferences 中保存的数据可以理解为 Map 型。我们通过 PreferenceManager 以及 getDefaultSharedPreferences(Context)来获取它，比如当我们想获得整数我们可以用 getInt(String key, int defVal) .获取里面的某个键值，当我们想修改时候我们用 putInt(String key, int newVal)， 最后用 edit()，方法提交!千万不要忘记了哦~

为了让大家跟好的理解我做了一个简单的 Demo,程序主要有个 TextView 控件，上面写着用户使用改应用的次数。效果如下图所示:



This app has been started 2 times.

下面是实现 Demo 的大体步骤:

一、新建一个 Android 工程命名为:PreferencesDemo。

二、在修改 main.xml 布局文件，这里只是在 TextView 控件里加了一个 id.代码如下:

view plaincopy to clipboardprint?

```
<?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″
      >
<TextView
      android:id=″@+id/text″
      android:layout_width=″fill_parent″
      android:layout_height=″wrap_content″
      android:text=″@string/hello″
      />
</LinearLayout>
```

```
<?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″
      >
<TextView
      android:id=″@+id/text″
      android:layout_width=″fill_parent″
      android:layout_height=″wrap_content″
      android:text=″@string/hello″
      />
</LinearLayout>
```

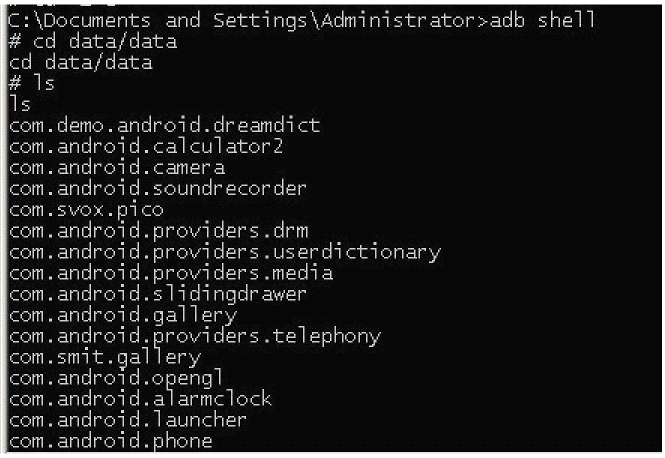
三、修改 PreferenceDemo.java 的代码，全部代码如下:

```
view plaincopy to clipboardprint?
package com.android.tutor;
import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.widget.TextView;
public class PreferencesDemo extends Activity {
      /** Called when the activity is first created. */
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);



              SharedPreferences mPerferences = PreferenceManager
              .getDefaultSharedPreferences(this);

              int counter = mPerferences.getInt("counter", 0);

              TextView mTextView = (TextView)findViewById(R.id.text);

              mTextView.setText("This app has been started " + counter + "
times.");

              SharedPreferences.Editor mEditor = mPerferences.edit();

              mEditor.putInt("counter", ++counter);
              mEditor.commit();
      }
}
package com.android.tutor;
import android.app.Activity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.widget.TextView;
public class PreferencesDemo extends Activity {
      /** Called when the activity is first created. */
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);



              SharedPreferences mPerferences = PreferenceManager
              .getDefaultSharedPreferences(this);

              int counter = mPerferences.getInt("counter", 0);

              TextView mTextView = (TextView)findViewById(R.id.text);

              mTextView.setText("This app has been started " + counter + " times.");



              SharedPreferences.Editor mEditor = mPerferences.edit();
```

```
            mEditor.putInt("counter", ++counter);
            mEditor.commit();


        }
}
```

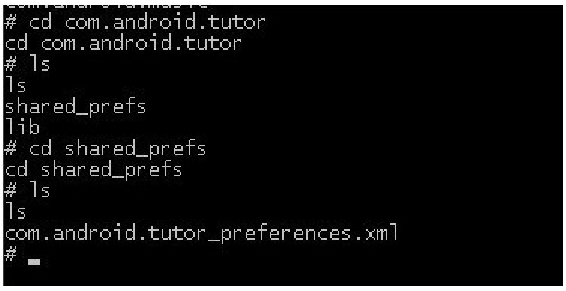
四、运行代码，实现上述效果.

五、查看 Preferences 文件，首先打开命令终端:adb shell 一下，然后 cd data/data 进入该目录，ls 一下我们会发现一大堆包文件，入下图所示:



```
C:\Documents and Settings\Administrator>adb shell
# cd data/data
cd data/data
# ls
ls
com.demo.android.dreamdict
com.android.calculator2
com.android.camera
com.android.soundrecorder
com.svox.pico
com.android.providers.drm
com.android.providers.userdictionary
com.android.providers.media
com.android.slidingdrawer
com.android.gallery
com.android.providers.telephony
com.smit.gallery
com.android.opengl
com.android.alarmclock
com.android.launcher
com.android.phone
```

cd com.android.tutor (这里是我程序的包名) /shared_prefs,ls 一下会发现.xml 文件如下图:



```
# cd com.android.tutor
cd com.android.tutor
# ls
ls
shared_prefs
lib
# cd shared_prefs
cd shared_prefs
# ls
ls
com.android.tutor_preferences.xml
# 
```

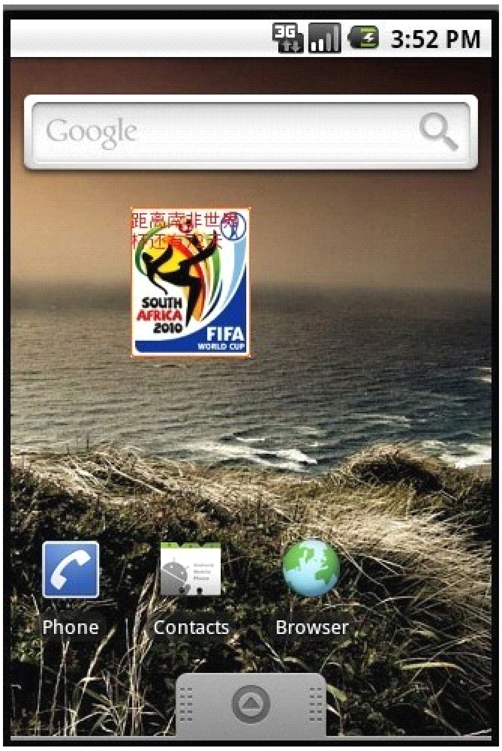
打开.xml 文件，格式如下(为什么这样大家自己去理解):

view plaincopy to clipboardprint?

```
<?xml version=’1.0’ encoding=’utf-8’ standalone=’yes’ ?>
<map>
<int name=″counter″ value=″3″ />
</map>
```

Android Widget 开发案例(世界杯倒计时!)

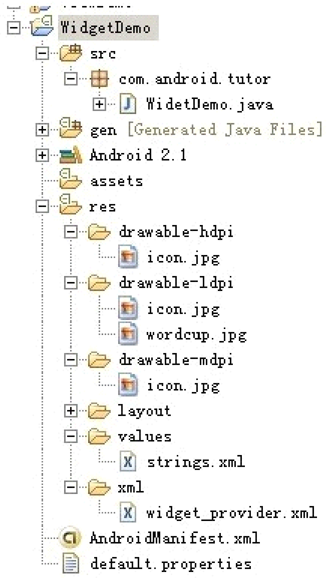
今天我们要写一下 Android Widget 的开发，由于快点凌晨，我就不说的太具体了，同志们就模仿吧！首先看一下效果图:



下面是 Demo 的详细步骤:

一、新建一个 Android 工程命名为:WidgetDemo.

二、准备素材，一个是 Widget 的图标，一个是 Widget 的背景。存放目录如下图:



```
WidgetDemo
  src
    com.android.tutor
      WidetDemo.java
  gen [Generated Java Files]
  Android 2.1
  assets
  res
    drawable-hdpi
      icon.jpg
    drawable-ldpi
      icon.jpg
      wordcup.jpg
    drawable-mdpi
      icon.jpg
    layout
    values
      strings.xml
    xml
      widget_provider.xml
  AndroidManifest.xml
  default.properties
```

三、修改 string.xml 文件如下:

view plaincopy to clipboardprint?

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

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

四、建立 Widget 内容提供者文件，我们在 res 下建立 xml 文件夹，并且新建一个 widget_provider.xml 代码入下:

```
view plaincopy to clipboardprint?
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
xmlns:android="http://schemas.android.com/apk/res/android"
      android:minWidth="50dip"
      android:minHeight="50dip"
      android:updatePeriodMillis="10000"
      android:initialLayout="@layout/main"
/>
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
android:minWidth="50dip"
android:minHeight="50dip"
android:updatePeriodMillis="10000"
android:initialLayout="@layout/main"
/>
```

五、修改 main.xml 布局，代码如下:

```
view plaincopy to clipboardprint?
<?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"
      android:background="@drawable/wordcup"
      >
<TextView
      android:id="@+id/wordcup"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
      android:textSize="12px"
      android:textColor="#ff0000"
      />
</LinearLayout>
<?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"
      android:background="@drawable/wordcup"
      >
<TextView
android:id="@+id/wordcup"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="@string/hello"
    android:textSize="12px"
      android:textColor="#ff0000"
      />
</LinearLayout>
```

六、修改 WidgetDemo.java 代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Timer;
import java.util.TimerTask;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.ComponentName;
import android.content.Context;
import android.widget.RemoteViews;
public class WidetDemo extends AppWidgetProvider {
      /** Called when the activity is first created. */

      @Override
                public  void  onUpdate(Context  context,  AppWidgetManager
appWidgetManager,
                    int[] appWidgetIds) {

              Timer timer = new Timer();
              timer.scheduleAtFixedRate(new MyTime(context,appWidgetManager), 1,
60000);
              super.onUpdate(context, appWidgetManager, appWidgetIds);
      }


      private class MyTime extends TimerTask{
              RemoteViews remoteViews;
              AppWidgetManager appWidgetManager;
              ComponentName thisWidget;

                              public  MyTime(Context  context,AppWidgetManager
appWidgetManager){
                    this.appWidgetManager = appWidgetManager;
```

```
                                                       remoteViews    =    new
RemoteViews(context.getPackageName(),R.layout.main);

                                                       thisWidget    =    new
ComponentName(context,WidetDemo.class);
            }
            public void run() {

                 Date date = new Date();
                 Calendar calendar = new GregorianCalendar(2010,06,11);
                                                           long    days    =
(((calendar.getTimeInMillis()-date.getTime())/1000))/86400;
                 remoteViews.setTextViewText(R.id.wordcup, "距离南非世界杯还
有" + days+"天");
                              appWidgetManager.updateAppWidget(thisWidget,
remoteViews);

            }

     }

}
package com.android.tutor;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
import java.util.Timer;
import java.util.TimerTask;
import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.ComponentName;
import android.content.Context;
import android.widget.RemoteViews;
public class WidetDemo extends AppWidgetProvider {
      /** Called when the activity is first created. */

@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
     int[] appWidgetIds) {

    Timer timer = new Timer();
    timer.scheduleAtFixedRate(new MyTime(context,appWidgetManager), 1, 60000);
    super.onUpdate(context, appWidgetManager, appWidgetIds);
}



private class MyTime extends TimerTask{
    RemoteViews remoteViews;
    AppWidgetManager appWidgetManager;
    ComponentName thisWidget;

    public MyTime(Context context,AppWidgetManager appWidgetManager){
     this.appWidgetManager = appWidgetManager;
     remoteViews = new RemoteViews(context.getPackageName(),R.layout.main);

     thisWidget = new ComponentName(context,WidetDemo.class);
    }
    public void run() {
     Date date = new Date();

     Calendar calendar = new GregorianCalendar(2010,06,11);

     long days = (((calendar.getTimeInMillis()-date.getTime())/1000))/86400;

     remoteViews.setTextViewText(R.id.wordcup, "距离南非世界杯还有" + days+"天");


     appWidgetManager.updateAppWidget(thisWidget, remoteViews);


    }


}


}
```

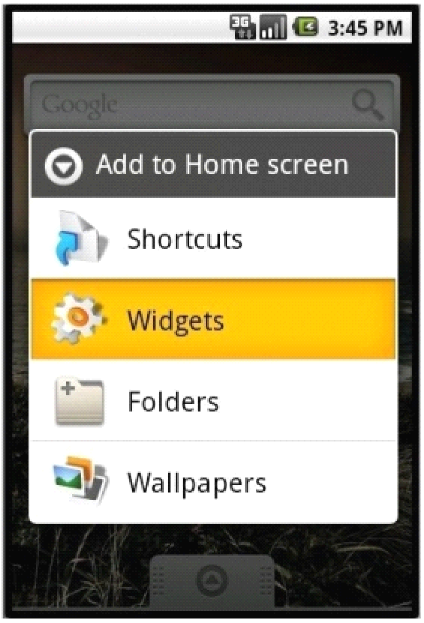
七、修改配置文件 AndroidManifest.xml，代码如下:

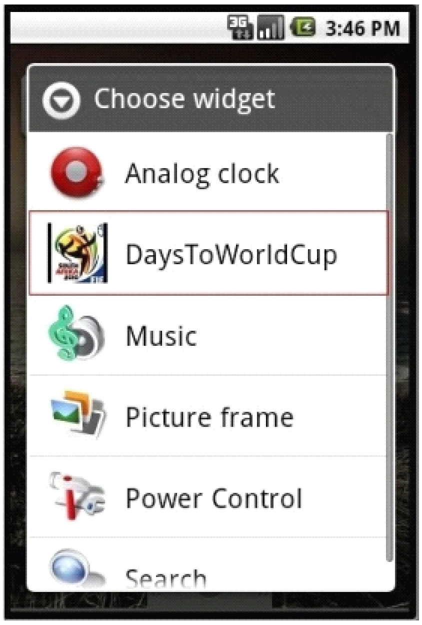
view plaincopy to clipboardprint?

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.android.tutor"
          android:versionCode="1"
          android:versionName="1.0">
                              <application       android:icon="@drawable/icon"
android:label="@string/app_name">
              <receiver android:name=".WidetDemo"
                              android:label="@string/app_name">
                    <intent-filter>
                                                                          <action
android:name="android.appwidget.action.APPWIDGET_UPDATE" />
                    </intent-filter>
                    <meta-data android:name="android.appwidget.provider"
```

```
                                        android:resource="@xml/widget_provider"

                    />
                </receiver>
        </application>
        <uses-sdk android:minSdkVersion="7" />
</manifest>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.android.tutor"
          android:versionCode="1"
          android:versionName="1.0">
                              <application       android:icon="@drawable/icon"
android:label="@string/app_name">
                <receiver android:name=".WidetDemo"
                              android:label="@string/app_name">
                    <intent-filter>
                                                                        <action
android:name="android.appwidget.action.APPWIDGET_UPDATE" />
                    </intent-filter>
                    <meta-data android:name="android.appwidget.provider"
                             android:resource="@xml/widget_provider"
                    />
                </receiver>
        </application>
        <uses-sdk android:minSdkVersion="7" />
</manifest>
```

八、点击运行(Ctrl+F11),之，运行成功后，我们长时间点击桌面，会出现如下俩个，依次点击，就可以看到最上面的效果图:


3:45 PM
Google
Add to Home screen
Shortcuts
Widgets
Folders
Wallpapers


3:46 PM
Choose widget
Analog clock
DaysToWorldCup
Music
Picture frame
Power Control
Search

Android Handler 的使用

大家好我们这一节讲的是 Android Handler 的使用,在讲 Handler 之前，我们先提个小问题，就是如何让程序 5 秒钟更新一下 Title.

首先我们看一下习惯了 Java 编程的人，在不知道 Handler 的用法之前是怎么样写的程序,代码如下所示:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.os.Bundle;
public class HandlerDemo extends Activity {

   //title 为 setTitle 方法提供变量,这里为了方便我设置成了 int 型
   private int title = 0;

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

      Timer timer = new Timer();
      timer.scheduleAtFixedRate(new MyTask(), 1, 5000);
   }

   private class MyTask extends TimerTask{
      @Override
```

```
       public void run() {

           setTitle("Welcome to Mr Wei's blog " + title);
           title ++;
       }
    }
}
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.os.Bundle;
public class HandlerDemo extends Activity {

//title 为 setTitle 方法提供变量,这里为了方便我设置成了 int 型
private int title = 0;

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

       Timer timer = new Timer();
       timer.scheduleAtFixedRate(new MyTask(), 1, 5000);
    }

    private class MyTask extends TimerTask{
 @Override
 public void run() {

  setTitle("Welcome to Mr Wei's blog " + title);
  title ++;
 }
    }
}
```

然而当我们执行程序，并不能达到我们预期的效果，所以 Android 引进了 Handler 这个特殊的类，可以说它是 Runnable 和 Activity 交互的桥梁

，所以我们只要在 run 方法中发送 Message,而在 Handler 里，通过不同的 Message 执行不同的任务。

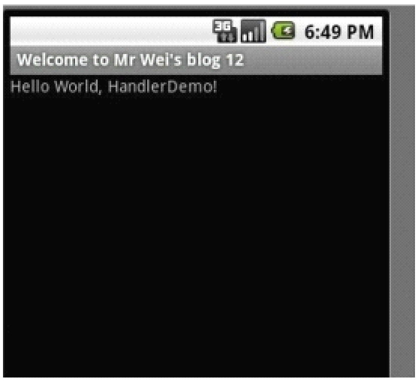
所以我们修改后的代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
public class HandlerDemo extends Activity {

    //title 为 setTitle 方法提供变量,这里为了方便我设置成了 int 型
    private int title = 0;

    private Handler mHandler = new Handler(){

        public void handleMessage(Message msg) {
            switch (msg.what) {
            case 1:
                updateTitle();
                break;
            }
        };
    };
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Timer timer = new Timer();
        timer.scheduleAtFixedRate(new MyTask(), 1, 5000);
    }

    private class MyTask extends TimerTask{
        @Override
        public void run() {

            Message message = new Message();
            message.what = 1;
            mHandler.sendMessage(message);

        }
    }


    public void updateTitle(){

        setTitle("Welcome to Mr Wei's blog " + title);
        title ++;
    }
}
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
public class HandlerDemo extends Activity {

//title 为 setTitle 方法提供变量,这里为了方便我设置成了 int 型
private int title = 0;

private Handler mHandler = new Handler(){

  public void handleMessage(Message msg) {
    switch (msg.what) {
    case 1:
      updateTitle();
      break;
    }
  };
};
      public void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.main);

          Timer timer = new Timer();
          timer.scheduleAtFixedRate(new MyTask(), 1, 5000);
      }

      private class MyTask extends TimerTask{
  @Override
```

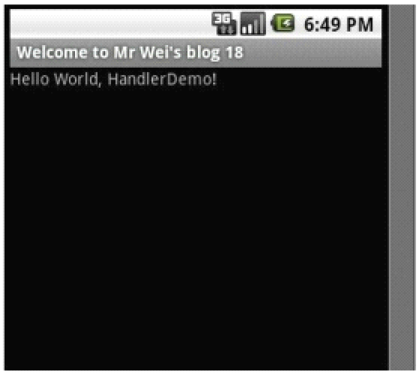
```
 public void run() {

   Message message = new Message();
   message.what = 1;
   mHandler.sendMessage(message);

 }
     }



     public void updateTitle(){

     setTitle("Welcome to Mr Wei's blog " + title);
 title ++;
     }
}
```

下面我们看一下效果图:



Welcome to Mr Wei's blog 12

Hello World, HandlerDemo!



Welcome to Mr Wei's blog 18

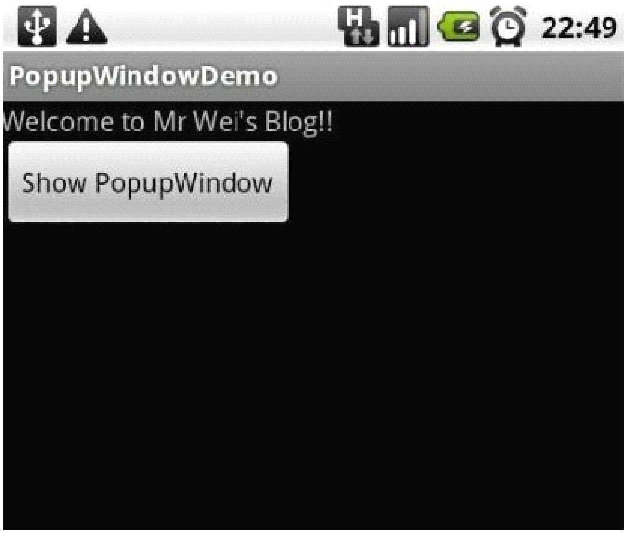
Hello World, HandlerDemo!

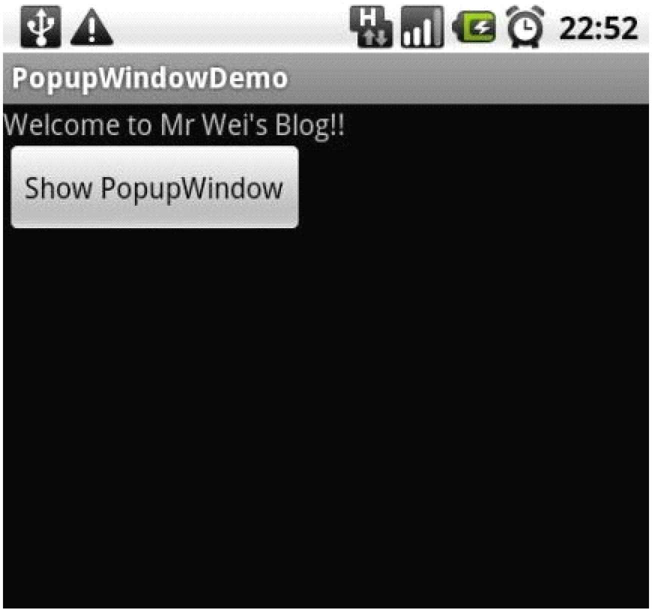
Android PopupWindow 的使用

大家好，我们这一节讲的是 Android PopupWindow 的使用! 在我理解其实 PopupWindow 其实类似于一个不能动的 Widget(仅从显示效果来说!)

它是浮在别的窗口之上的.

下面我将给大家做一个简单的 Demo，类似于音乐播放器的 Widget 的效果，点击 Button 的时候出来 PopupWindow,首先我们看一下效果图:





下面是核心代码:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.PopupWindow;
public class PopupWindowDemo extends Activity    implements OnClickListener{

      private Button btn;

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

              btn = (Button)findViewById(R.id.btn);
              btn.setOnClickListener(this);
      }
      @Override
      public void onClick(View v) {
              Context mContext = PopupWindowDemo.this;
              if (v.getId() == R.id.btn) {
                            LayoutInflater  mLayoutInflater  =  (LayoutInflater)
mContext
                                  .getSystemService(LAYOUT_INFLATER_SERVICE);

                    View music_popunwindwow = mLayoutInflater.inflate(
                                  R.layout.music_popwindow, null);
                    PopupWindow mPopupWindow = new PopupWindow(music_popunwindwow,
LayoutParams.FILL_PARENT,
                                  LayoutParams.WRAP_CONTENT);

                           mPopupWindow.showAtLocation(findViewById(R.id.main),
Gravity.RIGHT|Gravity.BOTTOM, 0, 0);
              }
      }
}
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.PopupWindow;
public class PopupWindowDemo extends Activity    implements OnClickListener{
private Button btn;

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

              btn = (Button)findViewById(R.id.btn);
              btn.setOnClickListener(this);
      }
@Override
public void onClick(View v) {
    Context mContext = PopupWindowDemo.this;
    if (v.getId() == R.id.btn) {
     LayoutInflater mLayoutInflater = (LayoutInflater) mContext
         .getSystemService(LAYOUT_INFLATER_SERVICE);
     View music_popunwindwow = mLayoutInflater.inflate(
         R.layout.music_popwindow, null);
          PopupWindow   mPopupWindow   =   new   PopupWindow(music_popunwindwow,
LayoutParams.FILL_PARENT,
         LayoutParams.WRAP_CONTENT);

                           mPopupWindow.showAtLocation(findViewById(R.id.main),
Gravity.RIGHT|Gravity.BOTTOM, 0, 0);
    }
}
}
```

需要强调的是这里 PopupWindow 必须有某个事件触发才会显示出来，不然总会抱错，不信大家可以试试!

随着这个问题的出现，就会同学问了，那么我想初始化让 PopupWindow 显示出来，那怎么办了，不去寄托于其他点击事件，

在这里我用了定时器 Timer 来实现这样的效果,当然这里就要用到 Handler 了，如果大家不理解的可以返回

Android Handler 的使用

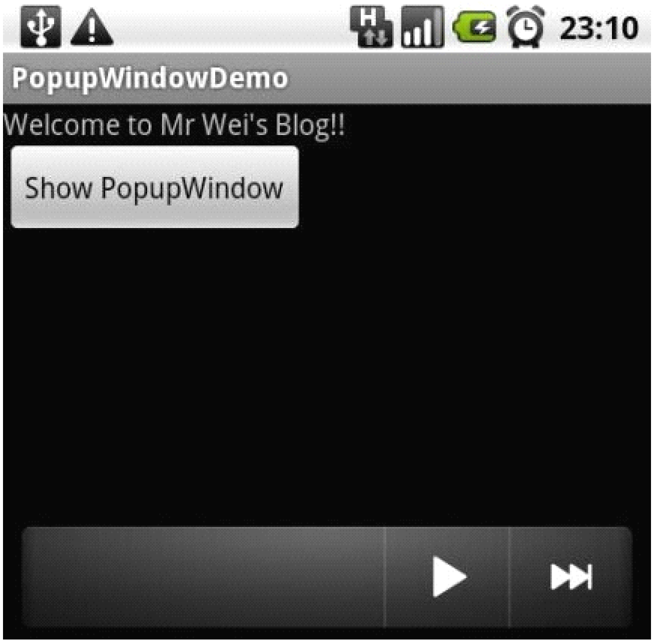
看一看，加深了解:

下面是核心代码:

```
view plaincopy to clipboardprint?
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.widget.PopupWindow;
public class PopupWindowDemo extends Activity{
      private Handler mHandler = new Handler(){

              public void handleMessage(Message msg) {
                    switch (msg.what) {
                    case 1:
                          showPopupWindow();
                          break;
                    }
              };
```

```
};

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

        //create the timer
        Timer timer = new Timer();
        timer.schedule(new initPopupWindow(), 100);
}

private class initPopupWindow extends TimerTask{
        @Override
        public void run() {

              Message message = new Message();
              message.what = 1;
              mHandler.sendMessage(message);

        }
}


public void showPopupWindow() {
        Context mContext = PopupWindowDemo.this;
        LayoutInflater mLayoutInflater = (LayoutInflater) mContext
                    .getSystemService(LAYOUT_INFLATER_SERVICE);
        View music_popunwindwow = mLayoutInflater.inflate(
                    R.layout.music_popwindow, null);
              PopupWindow mPopupWindow = new PopupWindow(music_popunwindwow,

                                                       LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
                           mPopupWindow.showAtLocation(findViewById(R.id.main),
Gravity.CENTER, 0, 0);
      }
}
package com.android.tutor;
import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup.LayoutParams;
import android.widget.PopupWindow;
public class PopupWindowDemo extends Activity{
private Handler mHandler = new Handler(){

    public void handleMessage(Message msg) {
     switch (msg.what) {
     case 1:
      showPopupWindow();
      break;
     }
    };
};

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

              //create the timer
              Timer timer = new Timer();
              timer.schedule(new initPopupWindow(), 100);
      }

      private class initPopupWindow extends TimerTask{
    @Override
    public void run() {

     Message message = new Message();
     message.what = 1;
     mHandler.sendMessage(message);

    }
      }



public void showPopupWindow() {
    Context mContext = PopupWindowDemo.this;
    LayoutInflater mLayoutInflater = (LayoutInflater) mContext
      .getSystemService(LAYOUT_INFLATER_SERVICE);
    View music_popunwindwow = mLayoutInflater.inflate(
      R.layout.music_popwindow, null);
    PopupWindow mPopupWindow = new PopupWindow(music_popunwindwow,
      LayoutParams.FILL_PARENT, LayoutParams.WRAP_CONTENT);
    mPopupWindow.showAtLocation(findViewById(R.id.main), Gravity.CENTER, 0, 0);
}
}
```

效果如下图:



Android 通用获取 Ip 的方法（判断手机是否联网的方法）

大家好，我们这一节讲一下，Android 获取 Ip 的一些方法,在我们开发中，有判断手机是否联网，或者想获得当前手机的 Ip 地址，当然 WIFI 连接的和我们 3G 卡的 Ip 地址当然是不一样的.

首先我尝试了如下方法:

view plaincopy to clipboardprint?

```
WifiManager wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);
WifiInfo wifiInfo = wifiManager.getConnectionInfo();
int ipAddress = wifiInfo.getIpAddress();
WifiManager wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);
WifiInfo wifiInfo = wifiManager.getConnectionInfo();
int ipAddress = wifiInfo.getIpAddress();
```

但是获得的居然是一个整数，我尝试了用些数学方法都没有成功!,所以这种方法不可取!

最后查了一些资料，发现如下方法是比较通用的，我尝试了 WIFI 和 G3 卡，都获取了争取的 Ip 地址:代码如下:

view plaincopy to clipboardprint?

```
public String getLocalIpAddress() {
    try {
         for (Enumeration<NetworkInterface> en = NetworkInterface.getNetworkInterfaces(); en.hasMoreElements();) {
        NetworkInterface intf = en.nextElement();
                  for (Enumeration<InetAddress> enumIpAddr = intf.getInetAddresses(); enumIpAddr.hasMoreElements();) {
```

```
           InetAddress inetAddress = enumIpAddr.nextElement();
           if (!inetAddress.isLoopbackAddress()) {
             return inetAddress.getHostAddress().toString();
           }
        }
     }
   } catch (SocketException ex) {
     Log.e(LOG_TAG, ex.toString());
   }
   return null;
}
public String getLocalIpAddress() {
   try {
         for (Enumeration<NetworkInterface> en = NetworkInterface.getNetworkInterfaces(); en.hasMoreElements();) {
        NetworkInterface intf = en.nextElement();
                   for (Enumeration<InetAddress> enumIpAddr = intf.getInetAddresses(); enumIpAddr.hasMoreElements();) {
           InetAddress inetAddress = enumIpAddr.nextElement();
           if (!inetAddress.isLoopbackAddress()) {
             return inetAddress.getHostAddress().toString();
           }
        }
     }
   } catch (SocketException ex) {
     Log.e(LOG_TAG, ex.toString());
   }
   return null;
}
```

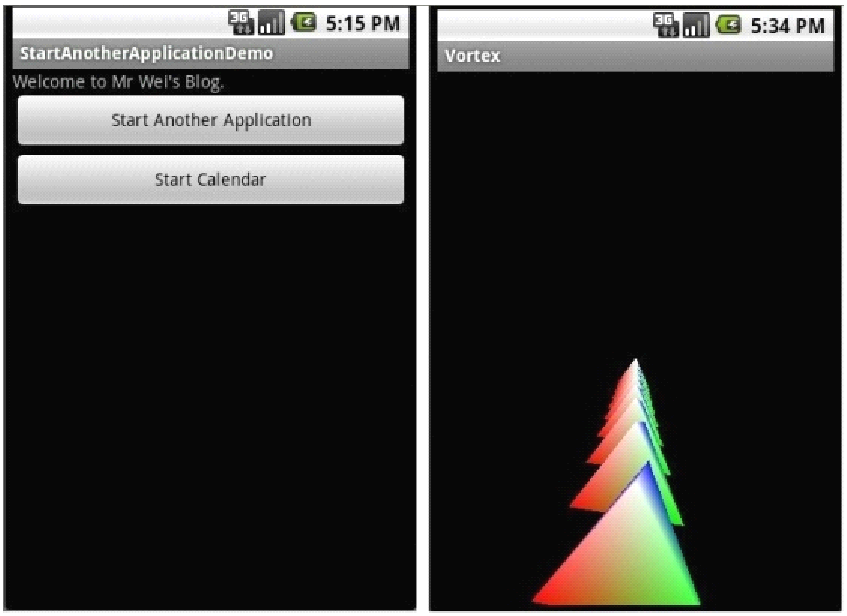
当我的手机处于飞行状态是，获得 Ip 地址为空，刚好符合要求!!!

Android 在一个应用中如何启动另外一个已安装的应用

今天晚上 Jimmy 问了我一个问题，就是如何在一个应用中 通过某个事件，而去启动另外一个已安装的应用。所以愿意和大家分享一下！

而为了能让大家更加容易的理解，我写了一个简单的 Demo，我们的程序有俩个按钮，其中一个点击会启动我自己写的应用(一个 3D 应用为例)，而另外一个按钮会启动系统自带的应用(如，日历，闹钟，计算器等等).这里我一日历为例子！

首先看一下我们的效果图(点击第一个按钮为例)：


5:15 PM
StartAnotherApplicationDemo
Welcome to Mr Wei's Blog.
Start Another Application
Start Calendar
5:34 PM
Vortex

下面是 Demo 的详细步骤：

一、新建一个 Android 工程命名为 StartAnotherApplicationDemo.

二、修改 main.xml 布局,代码如下：

view plaincopy to clipboardprint?

```
<?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"
      >
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Welcome to Mr Wei's Blog."
      />
<Button
      android:id="@+id/button"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Start Another Application"
/>
<Button
      android:id="@+id/start_calender"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Start Calendar"
/>
</LinearLayout>
<?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"
      >
```

```
<TextView
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="Welcome to Mr Wei's Blog."
      />
<Button
android:id="@+id/button"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Start Another Application"
/>
<Button
android:id="@+id/start_calender"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Start Calendar"
/>
</LinearLayout>
```

三、修改主程序 StartAnotherApplicationDemo.java 代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class StartAnotherApplicationDemo extends Activity {
      private Button mButton01,mButton02;

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

              mButton01 = (Button)findViewById(R.id.button);
              mButton02 = (Button)findViewById(R.id.start_calender);

              //-----启动我们自身写的程序------------------
              mButton01.setOnClickListener(new Button.OnClickListener(){
                    public void onClick(View v) {
                         //-----核心部分----- 前名一个参数是应用程序的包名,后
一个是这个应用程序的主 Activity 名
                          Intent intent=new Intent();
                                                            intent.setComponent(new
ComponentName("com.droidnova.android.games.vortex",

           "com.droidnova.android.games.vortex..Vortex"));
                          startActivity(intent);
                    }
              });
          //-----启动系统自带的应用程序------------------
              mButton02.setOnClickListener(new Button.OnClickListener(){
                    public void onClick(View v) {
                          Intent intent=new Intent();
                                                            intent.setComponent(new
ComponentName("com.android.calendar",
"com.android.calendar.LaunchActivity"));
                          startActivity(intent);
                    }
              });
      }
}
package com.android.tutor;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class StartAnotherApplicationDemo extends Activity {

private Button mButton01,mButton02;

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

              mButton01 = (Button)findViewById(R.id.button);
              mButton02 = (Button)findViewById(R.id.start_calender);

              //-----启动我们自身写的程序------------------
              mButton01.setOnClickListener(new Button.OnClickListener(){
     public void onClick(View v) {
      //-----核心部分----- 前名一个参数是应用程序的包名,后一个是这个应用程序的
主 Activity 名
```

```
      Intent intent=new Intent();
                                                     intent.setComponent(new
ComponentName("com.droidnova.android.games.vortex",
                                                        "com.droidnova.andro
id.games.vortex..Vortex"));
      startActivity(intent);
     }
             });
         //-----启动系统自带的应用程序------------------
             mButton02.setOnClickListener(new Button.OnClickListener(){
     public void onClick(View v) {
      Intent intent=new Intent();
               intent.setComponent(new   ComponentName("com.android.calendar",
"com.android.calendar.LaunchActivity"));
      startActivity(intent);
     }
             });
      }
}
```

四、执行之，将得到如上效果!

Android 数据库 SQLiteDatabase 的使用

大家好，好久没有更新博客了，最近由于身体不适让大家久等了，好了，直接进入主题~

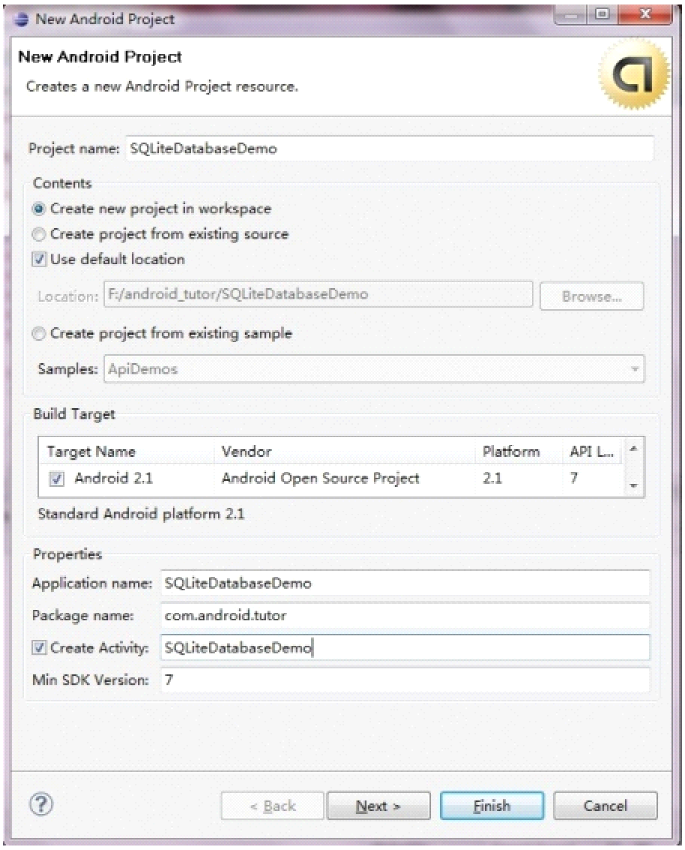
Android 提供了三种数据存储方式，第一种是文件存储;第二种是 SharedPreferences 存储;第三种就是数据库 SQLiteDatabase 存储。

文件存储我就不用多说了，而 SharedPreferences 可以存取简单的数据(int,double,float.etc)，它经常用于数据缓存，因为它读取存储简单。详细可以参见本系列。Android 高手进阶教程(七)之----Android 中 Preferences 的使用!

今天我们将讲一下 SQLiteDatabase 的使用。 而掌握 SqliteDatabase ，将会我们接下来掌握 ContentProvider 打下良好的基石。

为了让大家更好的掌握，我们手把手完成该节的 Demo 。

第一步：新建一个 Android 工程，命名为 SQLiteDatabaseDemo.



New Android Project

Creates a new Android Project resource.

Project name: SQLiteDatabaseDemo

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: F:/android_tutor/SQLiteDatabaseDemo

○ Create project from existing sample

Samples: ApiDemos

Build Target

Target Name	Vendor	Platform	API L...
☑ Android 2.1	Android Open Source Project	2.1	7

Standard Android platform 2.1

Properties

Application name: SQLiteDatabaseDemo

Package name: com.android.tutor

☑ Create Activity: SQLiteDatabaseDemo

Min SDK Version: 7

第二步:创建一个新的类 BooksDB.java 这个类要继承于

android.database.sqlite.SQLiteOpenHelper 抽象类，我们要实现其中两个方法：

onCreate(),onUpdate.具体代码如下：

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class BooksDB extends SQLiteOpenHelper {
          private final static String DATABASE_NAME = "BOOKS.db";
          private final static int DATABASE_VERSION = 1;
          private final static String TABLE_NAME = "books_table";
          public final static String BOOK_ID = "book_id";
          public final static String BOOK_NAME = "book_name";
          public final static String BOOK_AUTHOR = "book_author";

          public BooksDB(Context context) {
              // TODO Auto-generated constructor stub
                super(context, DATABASE_NAME, null, DATABASE_VERSION);
      }
      //创建 table
      @Override
      public void onCreate(SQLiteDatabase db) {
                String sql = "CREATE TABLE " + TABLE_NAME + " (" + BOOK_ID
                    + " INTEGER primary key autoincrement, " + BOOK_NAME + " text,
"+    BOOK_AUTHOR +" text);";
                db.execSQL(sql);
      }
      @Override
      public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
              String sql = "DROP TABLE IF EXISTS " + TABLE_NAME;
              db.execSQL(sql);
              onCreate(db);
      }
      public Cursor select() {
              SQLiteDatabase db = this.getReadableDatabase();
              Cursor cursor = db
                          .query(TABLE_NAME, null, null, null, null, null,
null);
              return cursor;
      }
      //增加操作
         public long insert(String bookname,String author)
          {
              SQLiteDatabase db = this.getWritableDatabase();
              /* ContentValues */
              ContentValues cv = new ContentValues();
              cv.put(BOOK_NAME, bookname);
              cv.put(BOOK_AUTHOR, author);
              long row = db.insert(TABLE_NAME, null, cv);
              return row;
          }
         //删除操作
          public void delete(int id)
          {
              SQLiteDatabase db = this.getWritableDatabase();
              String where = BOOK_ID + " = ?";
              String[] whereValue ={ Integer.toString(id) };
              db.delete(TABLE_NAME, where, whereValue);
          }
          //修改操作
          public void update(int id, String bookname,String author)
          {
              SQLiteDatabase db = this.getWritableDatabase();
              String where = BOOK_ID + " = ?";
              String[] whereValue = { Integer.toString(id) };

              ContentValues cv = new ContentValues();
              cv.put(BOOK_NAME, bookname);
              cv.put(BOOK_AUTHOR, author);
              db.update(TABLE_NAME, cv, where, whereValue);
          }
}
package com.android.tutor;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class BooksDB extends SQLiteOpenHelper {
     private final static String DATABASE_NAME = "BOOKS.db";
     private final static int DATABASE_VERSION = 1;
     private final static String TABLE_NAME = "books_table";
     public final static String BOOK_ID = "book_id";
     public final static String BOOK_NAME = "book_name";
     public final static String BOOK_AUTHOR = "book_author";

     public BooksDB(Context context) {
    // TODO Auto-generated constructor stub
      super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
//创建 table
```

```
@Override
public void onCreate(SQLiteDatabase db) {
      String sql = "CREATE TABLE " + TABLE_NAME + " (" + BOOK_ID
               + " INTEGER primary key autoincrement, " + BOOK_NAME + " text,
"+    BOOK_AUTHOR +" text);";
           db.execSQL(sql);
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    String sql = "DROP TABLE IF EXISTS " + TABLE_NAME;
    db.execSQL(sql);
    onCreate(db);
}

public Cursor select() {
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db
      .query(TABLE_NAME, null, null, null, null, null, null);
    return cursor;
}
//增加操作
    public long insert(String bookname,String author)
     {
         SQLiteDatabase db = this.getWritableDatabase();
         /* ContentValues */
         ContentValues cv = new ContentValues();
         cv.put(BOOK_NAME, bookname);
         cv.put(BOOK_AUTHOR, author);
         long row = db.insert(TABLE_NAME, null, cv);
```

```
        return row;
    }
   //删除操作
    public void delete(int id)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        String where = BOOK_ID + ″ = ?″;
        String[] whereValue ={ Integer.toString(id) };
        db.delete(TABLE_NAME, where, whereValue);
    }
    //修改操作
    public void update(int id, String bookname,String author)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        String where = BOOK_ID + ″ = ?″;
        String[] whereValue = { Integer.toString(id) };

        ContentValues cv = new ContentValues();
        cv.put(BOOK_NAME, bookname);
        cv.put(BOOK_AUTHOR, author);
        db.update(TABLE_NAME, cv, where, whereValue);
    }
}
```

第三步:修改 main.xml 布局如下，由两个 EditText 和一个 ListView 组成，代码如下:

view plaincopy to clipboardprint?

```
<?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/bookname"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          >
          </EditText>
          <EditText
          android:id="@+id/author"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          >
          </EditText>
          <ListView
          android:id="@+id/bookslist"
          android:layout_width="fill_parent"
          android:layout_height="wrap_content"
          >
          </ListView>
</LinearLayout>
<?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/bookname"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     >
     </EditText>
     <EditText
     android:id="@+id/author"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     >
     </EditText>
     <ListView
     android:id="@+id/bookslist"
     android:layout_width="fill_parent"
     android:layout_height="wrap_content"
     >
     </ListView>
</LinearLayout>
```

第四步:修改 SQLiteDatabaseDemo.java 代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.database.Cursor;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
public class SQLiteDatabaseDemo extends Activity implements
AdapterView.OnItemClickListener {
      private BooksDB    mBooksDB;
      private Cursor     mCursor;
      private EditText BookName;
      private EditText BookAuthor;
      private ListView BooksList;

      private int BOOK_ID = 0;
      protected final static int MENU_ADD = Menu.FIRST;
      protected final static int MENU_DELETE = Menu.FIRST + 1;
      protected final static int MENU_UPDATE = Menu.FIRST + 2;

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

      public void setUpViews(){
              mBooksDB = new BooksDB(this);
```

```
        mCursor    = mBooksDB.select();

        BookName = (EditText)findViewById(R.id.bookname);
        BookAuthor = (EditText)findViewById(R.id.author);
        BooksList = (ListView)findViewById(R.id.bookslist);

        BooksList.setAdapter(new BooksListAdapter(this, mCursor));
        BooksList.setOnItemClickListener(this);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
        super.onCreateOptionsMenu(menu);

        menu.add(Menu.NONE, MENU_ADD, 0, "ADD");
        menu.add(Menu.NONE, MENU_DELETE, 0, "DELETE");
        menu.add(Menu.NONE, MENU_DELETE, 0, "UPDATE");
        return true;
}

public boolean onOptionsItemSelected(MenuItem item)
{
    super.onOptionsItemSelected(item);
    switch (item.getItemId())
    {
        case MENU_ADD:
              add();
              break;
        case MENU_DELETE:
```

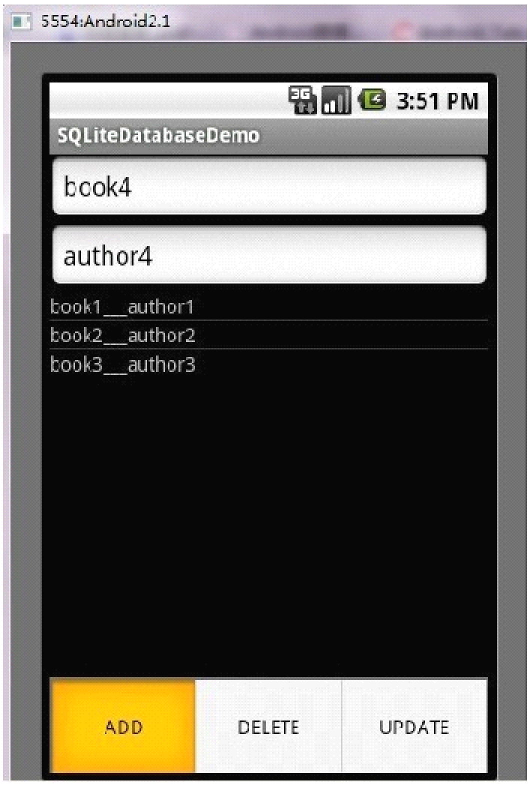
```
                    delete();
                    break;
              case MENU_UPDATE:
                    update();
                    break;
          }
          return true;
      }

      public void add(){
              String bookname = BookName.getText().toString();
              String author    = BookAuthor.getText().toString();
              //书名和作者都不能为空，或者退出
              if (bookname.equals("") || author.equals("")){
                    return;
              }
               mBooksDB.insert(bookname, author);
               mCursor.requery();
               BooksList.invalidateViews();
               BookName.setText("");
               BookAuthor.setText("");
               Toast.makeText(this, "Add Successed!",
Toast.LENGTH_SHORT).show();
      }

      public void delete(){
              if (BOOK_ID == 0) {
                    return;
              }
```

```
        mBooksDB.delete(BOOK_ID);
        mCursor.requery();
        BooksList.invalidateViews();
        BookName.setText("");
        BookAuthor.setText("");
        Toast.makeText(this, "Delete Successed!",
Toast.LENGTH_SHORT).show();
    }

    public void update(){
        String bookname = BookName.getText().toString();
        String author    = BookAuthor.getText().toString();
        //书名和作者都不能为空，或者退出
        if (bookname.equals("") || author.equals("")){
            return;
        }
        mBooksDB.update(BOOK_ID, bookname, author);
        mCursor.requery();
        BooksList.invalidateViews();
        BookName.setText("");
        BookAuthor.setText("");
        Toast.makeText(this, "Update Successed!",
Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        mCursor.moveToPosition(position);
        BOOK_ID = mCursor.getInt(0);
        BookName.setText(mCursor.getString(1));
        BookAuthor.setText(mCursor.getString(2));

}

public class BooksListAdapter extends BaseAdapter{
        private Context mContext;
        private Cursor mCursor;
        public BooksListAdapter(Context context,Cursor cursor) {

              mContext = context;
              mCursor = cursor;
        }
        @Override
        public int getCount() {
              return mCursor.getCount();
        }
        @Override
        public Object getItem(int position) {
              return null;
        }
        @Override
        public long getItemId(int position) {
              return 0;
        }
        @Override
        public View getView(int position, View convertView, ViewGroup parent)
```

```
{
                    TextView mTextView = new TextView(mContext);
                    mCursor.moveToPosition(position);
                    mTextView.setText(mCursor.getString(1) + "___" +
mCursor.getString(2));
                    return mTextView;
              }


      }
}
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.database.Cursor;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;
public class SQLiteDatabaseDemo extends Activity implements
AdapterView.OnItemClickListener {
private BooksDB    mBooksDB;
private Cursor     mCursor;
private EditText BookName;
private EditText BookAuthor;
private ListView BooksList;

private int BOOK_ID = 0;
protected final static int MENU_ADD = Menu.FIRST;
protected final static int MENU_DELETE = Menu.FIRST + 1;
protected final static int MENU_UPDATE = Menu.FIRST + 2;

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

      public void setUpViews(){
         mBooksDB = new BooksDB(this);
         mCursor    = mBooksDB.select();

         BookName = (EditText)findViewById(R.id.bookname);
         BookAuthor = (EditText)findViewById(R.id.author);
         BooksList = (ListView)findViewById(R.id.bookslist);

         BooksList.setAdapter(new BooksListAdapter(this, mCursor));
         BooksList.setOnItemClickListener(this);
      }

      @Override
      public boolean onCreateOptionsMenu(Menu menu) {
```

```
     super.onCreateOptionsMenu(menu);

          menu.add(Menu.NONE, MENU_ADD, 0, "ADD");
          menu.add(Menu.NONE, MENU_DELETE, 0, "DELETE");
          menu.add(Menu.NONE, MENU_DELETE, 0, "UPDATE");
          return true;
  }

  public boolean onOptionsItemSelected(MenuItem item)
  {
      super.onOptionsItemSelected(item);
      switch (item.getItemId())
      {
case MENU_ADD:
 add();
 break;
case MENU_DELETE:
 delete();
 break;
case MENU_UPDATE:
 update();
 break;
      }
      return true;
  }

  public void add(){
     String bookname = BookName.getText().toString();
     String author    = BookAuthor.getText().toString();
     //书名和作者都不能为空，或者退出
     if (bookname.equals("") || author.equals("")){
      return;
     }
      mBooksDB.insert(bookname, author);
      mCursor.requery();
      BooksList.invalidateViews();
      BookName.setText("");
      BookAuthor.setText("");
      Toast.makeText(this, "Add Successed!", Toast.LENGTH_SHORT).show();
  }

  public void delete(){
if (BOOK_ID == 0) {
 return;
}
mBooksDB.delete(BOOK_ID);
mCursor.requery();
BooksList.invalidateViews();
BookName.setText("");
BookAuthor.setText("");
Toast.makeText(this, "Delete Successed!", Toast.LENGTH_SHORT).show();
  }

  public void update(){
     String bookname = BookName.getText().toString();
     String author    = BookAuthor.getText().toString();
     //书名和作者都不能为空，或者退出
     if (bookname.equals("") || author.equals("")){
          return;
         }
         mBooksDB.update(BOOK_ID, bookname, author);
         mCursor.requery();
    BooksList.invalidateViews();
    BookName.setText("");
    BookAuthor.setText("");
    Toast.makeText(this, "Update Successed!", Toast.LENGTH_SHORT).show();
      }

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id)
{

    mCursor.moveToPosition(position);
    BOOK_ID = mCursor.getInt(0);
    BookName.setText(mCursor.getString(1));
    BookAuthor.setText(mCursor.getString(2));

}

      public class BooksListAdapter extends BaseAdapter{
         private Context mContext;
         private Cursor mCursor;
         public BooksListAdapter(Context context,Cursor cursor) {

          mContext = context;
          mCursor = cursor;
    }
    @Override
    public int getCount() {
     return mCursor.getCount();
    }
    @Override
    public Object getItem(int position) {
     return null;
    }
    @Override
    public long getItemId(int position) {
     return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
     TextView mTextView = new TextView(mContext);
     mCursor.moveToPosition(position);
     mTextView.setText(mCursor.getString(1) + "___" + mCursor.getString(2));
     return mTextView;
    }

      }
}
```

第五步:运行程序效果如下:



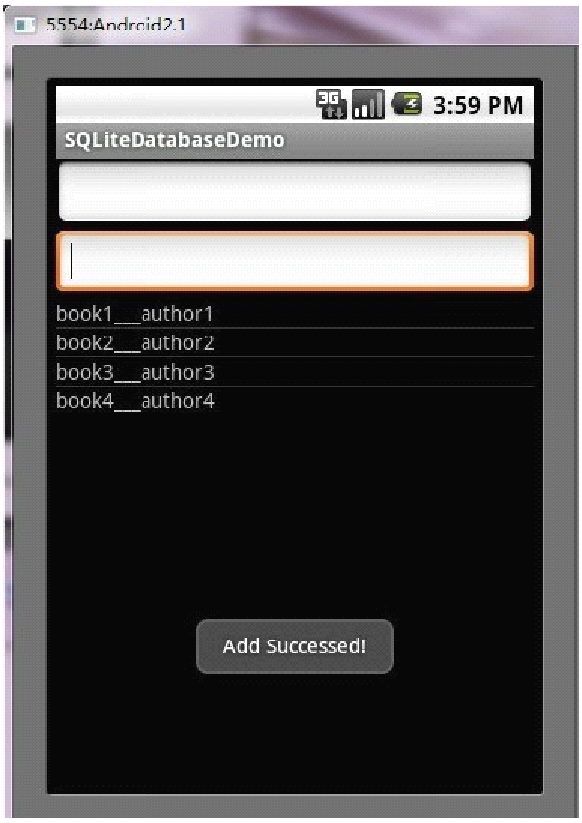
book4

author4

book1__author1

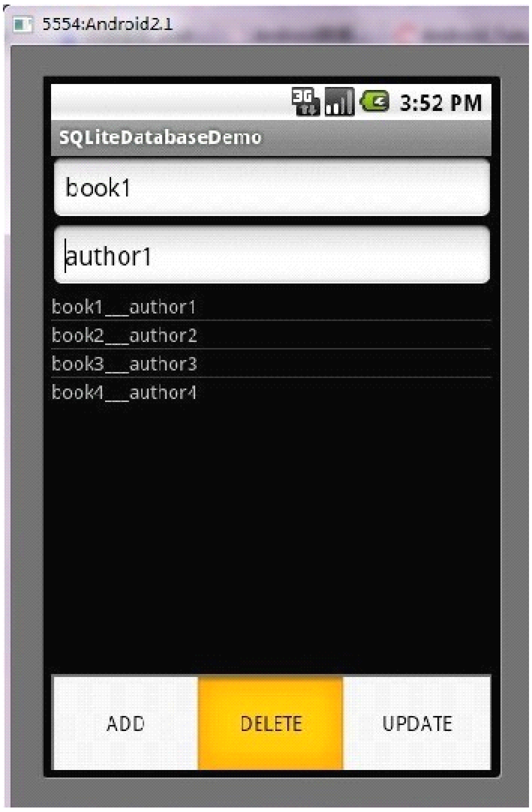
book2__author2

book3__author3



```
book1___author1
book2___author2
book3___author3
book4___author4
```

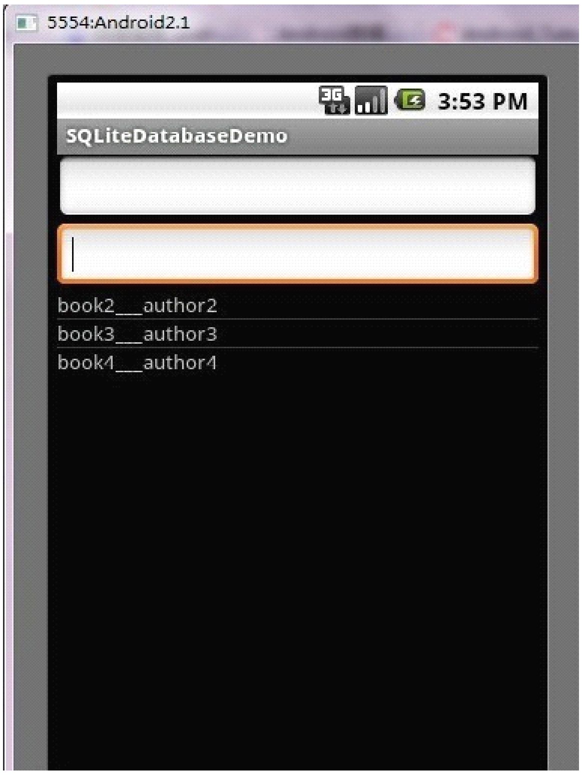
Add Successed!



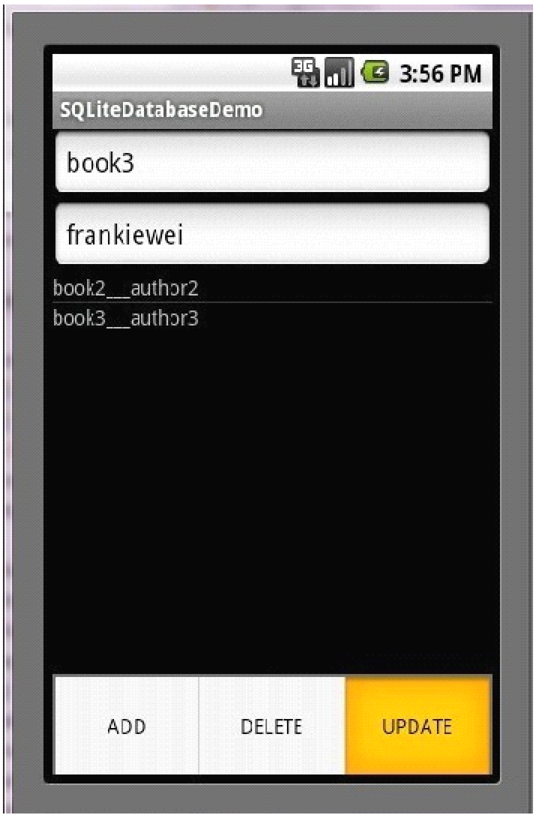
book1

author1

```
book1__author1
book2__author2
book3__author3
book4__author4
```



book2___author2
book3___author3
book4___author4

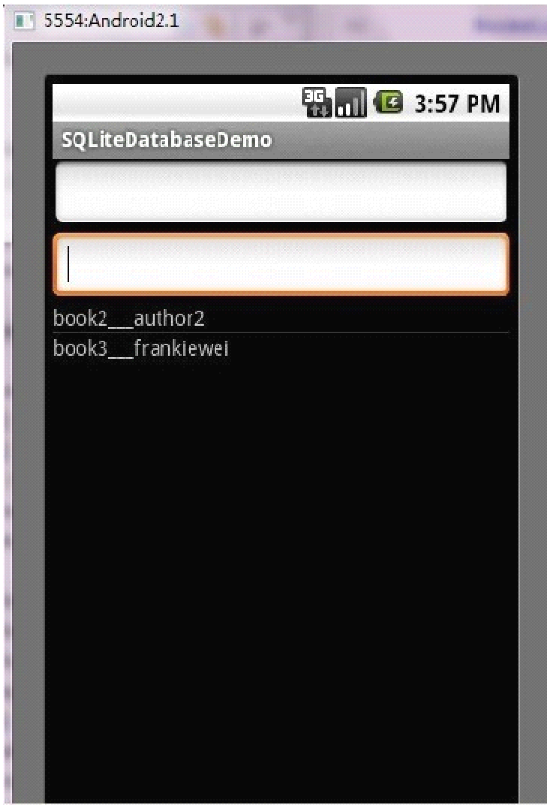


book3

frankiewei

book2___author2

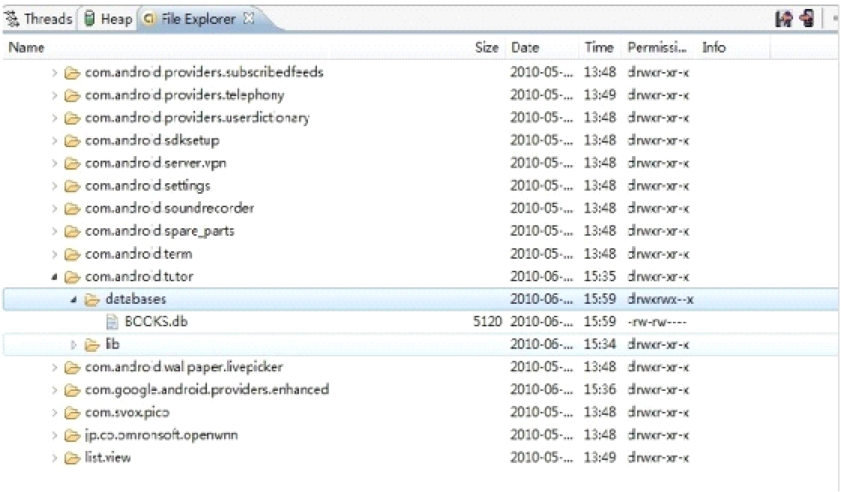
book3___author3



book2__author2
book3__frankiewei

第六步：查看我们所建的数据库。有两种方法：第一种用命令查看：adb shell ls data/data/com.android.tutor/databases。

另一种方法是用 DDMS 查看，在 data/data 下面对应的应用程序的包名 下会有如下数据库，如图所示：



Name	Size	Date	Time	Permissi...	Info
com.android.providers.subscribedfeeds		2010-05-...	13:48	drwxr-xr-x	
com.android.providers.telephony		2010-05-...	13:49	drwxr-xr-x	
com.android.providers.userdictionary		2010-05-...	13:48	drwxr-xr-x	
com.android.sdksetup		2010-05-...	13:48	drwxr-xr-x	
com.android.server.vpn		2010-05-...	13:48	drwxr-xr-x	
com.android.settings		2010-05-...	13:48	drwxr-xr-x	
com.android.soundrecorder		2010-05-...	13:48	drwxr-xr-x	
com.android.spare_parts		2010-05-...	13:48	drwxr-xr-x	
com.android.term		2010-05-...	13:48	drwxr-xr-x	
com.android.tutor		2010-06-...	15:35	drwxr-xr-x	
databases		2010-06-...	15:59	drwxrwx--x	
BOOKS.db	5120	2010-06-...	15:59	-rw-rw----	
lib		2010-06-...	15:34	drwxr-xr-x	
com.android.wallpaper.livepicker		2010-05-...	13:48	drwxr-xr-x	
com.google.android.providers.enhanced		2010-06-...	15:36	drwxr-xr-x	
com.svox.pico		2010-05-...	13:48	drwxr-xr-x	
jp.co.omronsoft.openwnn		2010-05-...	13:48	drwxr-xr-x	
list.view		2010-05-...	13:49	drwxr-xr-x	

Android Location 的使用

大家好，今天说说 Location , Location 在 Android 开发中还是经常用到的，比如 通过经纬度获取天气，根据 Location 获取所在地区详细 Address （比如 Google Map 开发).等。而在 Android 中通过 LocationManager 来获取 Location .通常获取 Location 有 GPS 获取，WIFI 获取。

我今天做一个简单的小 Demo ,来教大家如何获取 Location ,从而获取经纬度。下一节将教大家通过 Location 来获取 Address .

首先第一步:

创建一个 Android 工程命名为 LocationDemo .

第二步：修改 main.xml 代码如下:

view plaincopy to clipboardprint?

```
<?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"
    >
<TextView
    android:id="@+id/longitude"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="longitude:"
    />
<TextView
    android:id="@+id/latitude"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="latitude:"
    />
</LinearLayout>
<?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"
    >
<TextView
android:id="@+id/longitude"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="longitude:"
    />
<TextView
android:id="@+id/latitude"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="latitude:"
    />
</LinearLayout>
```

第三步:修改 LocationDemo.java ，代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
public class LocationDemo extends Activity {

    private TextView longitude;
    private TextView latitude;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        longitude = (TextView)findViewById(R.id.longitude);
        latitude = (TextView)findViewById(R.id.latitude);

        Location mLocation = getLocation(this);

        longitude.setText("Longitude: " + mLocation.getLongitude());
        latitude.setText("Latitude: " + mLocation.getLatitude());
    }

    //Get the Location by GPS or WIFI
    public Location getLocation(Context context) {
        LocationManager locMan = (LocationManager) context
                .getSystemService(Context.LOCATION_SERVICE);
        Location location = locMan
                .getLastKnownLocation(LocationManager.GPS_PROVIDER);
        if (location == null) {
            location = locMan
                    .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
        }
        return location;
    }
}
package com.android.tutor;
import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
public class LocationDemo extends Activity {

private TextView longitude;
private TextView latitude;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        longitude = (TextView)findViewById(R.id.longitude);
        latitude = (TextView)findViewById(R.id.latitude);

        Location mLocation = getLocation(this);

        longitude.setText("Longitude: " + mLocation.getLongitude());
        latitude.setText("Latitude: " + mLocation.getLatitude());
    }

    //Get the Location by GPS or WIFI
```

```
public Location getLocation(Context context) {
  LocationManager locMan = (LocationManager) context
      .getSystemService(Context.LOCATION_SERVICE);
  Location location = locMan
      .getLastKnownLocation(LocationManager.GPS_PROVIDER);
  if (location == null) {
    location = locMan
      .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
  }
  return location;
}
}
```

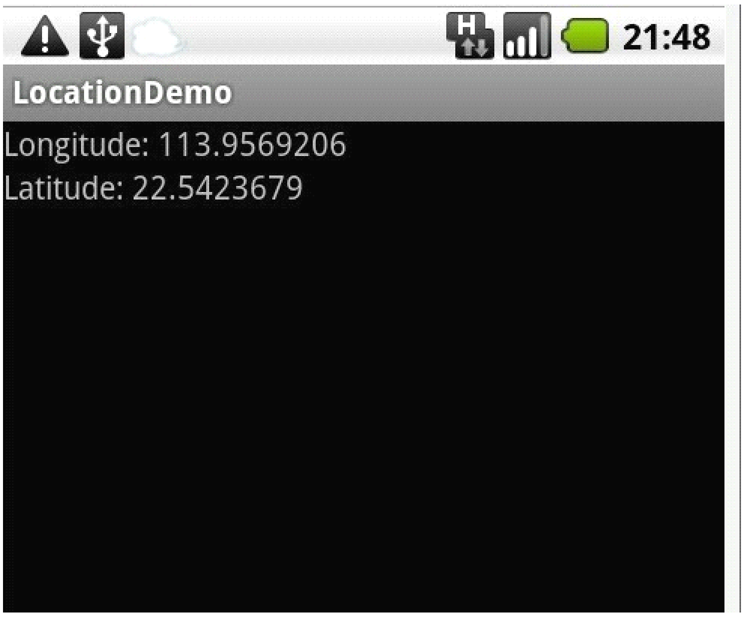
第四步：增加权限，修改 AndroidManifest.xml 代码如下（第 16 行为所增行）：

view plaincopy to clipboardprint?

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
        package="com.android.tutor"
        android:versionCode="1"
        android:versionName="1.0">
      <application android:icon="@drawable/icon" android:label="@string/app_name">
          <activity android:name=".LocationDemo"
                    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="7" />
      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

```
</manifest>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.android.tutor"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".LocationDemo"
                  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="7" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
</manifest>
```

第五步:运行 LocationDemo 工程，所得效果如下(真机深圳测试):



```
Longitude: 113.9569206
Latitude: 22.5423679
```

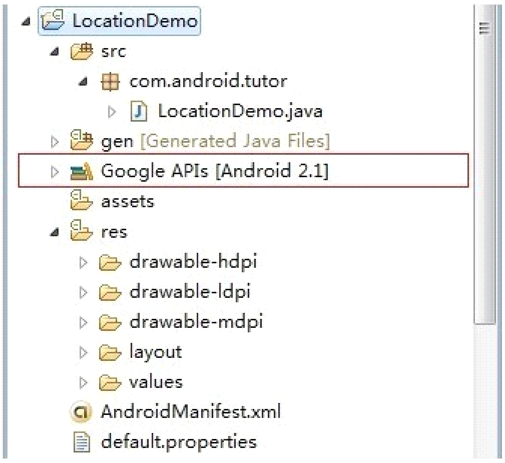
通过 Location 获取 Address 的使用

大家好，上一节我讲了一下如何通过 LocationManager 来获取 Location，没有看过上一节的同学，可以点击如下链接返回查看：

Android 高手进阶教程十四之---Android Location 的使用!

我们获取 Location 的目的之一肯定是有获取这个位置的详细地址，而我们有了 Location 在来获取 Address 就相对简单多了，因为 GoogleApi 已经封装好了方法，我们只需呀通过 Location 获取 GeoPoint,然后在通过 GeoPoint 来获取我们想要的 Address.下面是我做的一个简单的 Demo.

第一步新建一个 Android 工程 LocationDemo，注意这里选用的是(Google APIs)，下面是文件目录结构：



```
LocationDemo
  src
    com.android.tutor
      LocationDemo.java
  gen [Generated Java Files]
  Google APIs [Android 2.1]
  assets
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
    values
  AndroidManifest.xml
  default.properties
```

第二步： 修改 main.xml(相比第十四节增加了一个 address 的 TextView)，代码如下:

view plaincopy to clipboardprint?

```
<?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"
	>
<TextView
	android:id="@+id/longitude"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content"
	android:text="longitude:"
	/>
<TextView
	android:id="@+id/latitude"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content"
	android:text="latitude:"
	/>
<TextView
	android:id="@+id/address"
	android:layout_width="fill_parent"
	android:layout_height="wrap_content"
	/>
</LinearLayout>
<?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"
	>
<TextView
android:id="@+id/longitude"
```

```
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="longitude:"
      />
<TextView
android:id="@+id/latitude"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:text="latitude:"
      />
<TextView
android:id="@+id/address"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      />
</LinearLayout>
```

第三步:修改 LocationDemo.java(增加了两个方法)代码如下:

view plaincopy to clipboardprint?

```
package com.android.tutor;
import java.util.List;
import java.util.Locale;
import com.google.android.maps.GeoPoint;
import android.app.Activity;
import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
public class LocationDemo extends Activity {

      private TextView longitude;
      private TextView latitude;
      private TextView address;
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);

              longitude = (TextView)findViewById(R.id.longitude);
              latitude = (TextView)findViewById(R.id.latitude);
              address = (TextView)findViewById(R.id.address);

              Location mLocation = getLocation(this);
              GeoPoint gp = getGeoByLocation(mLocation);
              Address mAddress = getAddressbyGeoPoint(this, gp);




              longitude.setText("Longitude: " + mLocation.getLongitude());
              latitude.setText("Latitude: " + mLocation.getLatitude());
                 address.setText("Address: " + mAddress.getCountryName()+"," +
mAddress.getLocality());
      }

      //Get the Location by GPS or WIFI
      public Location getLocation(Context context) {
              LocationManager locMan = (LocationManager) context
                          .getSystemService(Context.LOCATION_SERVICE);
              Location location = locMan
                           .getLastKnownLocation(LocationManager.GPS_PROVIDER);

              if (location == null) {
                    location = locMan
                                  .getLastKnownLocation(LocationManager.NETWORK
_PROVIDER);
              }
              return location;
      }
      //通过 Location 获取 GeoPoint
         public    GeoPoint getGeoByLocation(Location location) {
               GeoPoint gp = null;
               try {
                     if (location != null) {
                                 double  geoLatitude  =  location.getLatitude()  *
1E6;
                               double  geoLongitude  =  location.getLongitude()  *
1E6;
                                   gp  =  new  GeoPoint((int)  geoLatitude,  (int)
geoLongitude);
                     }
               } catch (Exception e) {
                     e.printStackTrace();
               }
               return gp;
         }
```

```
        //通过 GeoPoint 来获取 Address
         public    Address getAddressbyGeoPoint(Context cntext, GeoPoint gp)
{
              Address result = null;
              try {
                    if (gp != null) {
                                         Geocoder gc = new Geocoder(cntext,
Locale.CHINA);

                                double geoLatitude = (int) gp.getLatitudeE6() /
1E6;
                              double geoLongitude = (int) gp.getLongitudeE6() /
1E6;

                                                 List<Address>  lstAddress  =
gc.getFromLocation(geoLatitude,
                                          geoLongitude, 1);
                             if (lstAddress.size() > 0) {
                                   result = lstAddress.get(0);
                             }
                    }
              } catch (Exception e) {
                    e.printStackTrace();
              }
              return result;
        }
}
package com.android.tutor;
import java.util.List;
import java.util.Locale;
import com.google.android.maps.GeoPoint;
import android.app.Activity;
import android.content.Context;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
public class LocationDemo extends Activity {

private TextView longitude;
private TextView latitude;
private TextView address;
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);

              longitude = (TextView)findViewById(R.id.longitude);
              latitude = (TextView)findViewById(R.id.latitude);
              address = (TextView)findViewById(R.id.address);

              Location mLocation = getLocation(this);
              GeoPoint gp = getGeoByLocation(mLocation);
              Address mAddress = getAddressbyGeoPoint(this, gp);
              longitude.setText("Longitude: " + mLocation.getLongitude());
              latitude.setText("Latitude: " + mLocation.getLatitude());
                address.setText("Address: " + mAddress.getCountryName()+"," +
mAddress.getLocality());
      }

      //Get the Location by GPS or WIFI
public Location getLocation(Context context) {
    LocationManager locMan = (LocationManager) context
      .getSystemService(Context.LOCATION_SERVICE);
    Location location = locMan
      .getLastKnownLocation(LocationManager.GPS_PROVIDER);
    if (location == null) {
     location = locMan
         .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
    }
    return location;
}
//通过 Location 获取 GeoPoint
    public  GeoPoint getGeoByLocation(Location location) {
          GeoPoint gp = null;
          try {
                if (location != null) {
                        double geoLatitude = location.getLatitude() * 1E6;
                        double geoLongitude = location.getLongitude() * 1E6;
                        gp = new GeoPoint((int) geoLatitude, (int) geoLongitude);

                }
          } catch (Exception e) {
                e.printStackTrace();
          }
          return gp;
    }
    //通过 GeoPoint 来获取 Address
    public    Address getAddressbyGeoPoint(Context cntext, GeoPoint gp) {
          Address result = null;
          try {
                if (gp != null) {
                        Geocoder gc = new Geocoder(cntext, Locale.CHINA);

                        double geoLatitude = (int) gp.getLatitudeE6() / 1E6;
                        double geoLongitude = (int) gp.getLongitudeE6() / 1E6;

                                                  List<Address>   lstAddress   =
gc.getFromLocation(geoLatitude,
                                    geoLongitude, 1);
                        if (lstAddress.size() > 0) {
                              result = lstAddress.get(0);
                        }
                }
          } catch (Exception e) {
                e.printStackTrace();
          }
          return result;
    }
}
```

第四步:最重要一步在 AndroidManiefest.xml 中导入 Google Api(第 14 行代码)库,代码如

下:

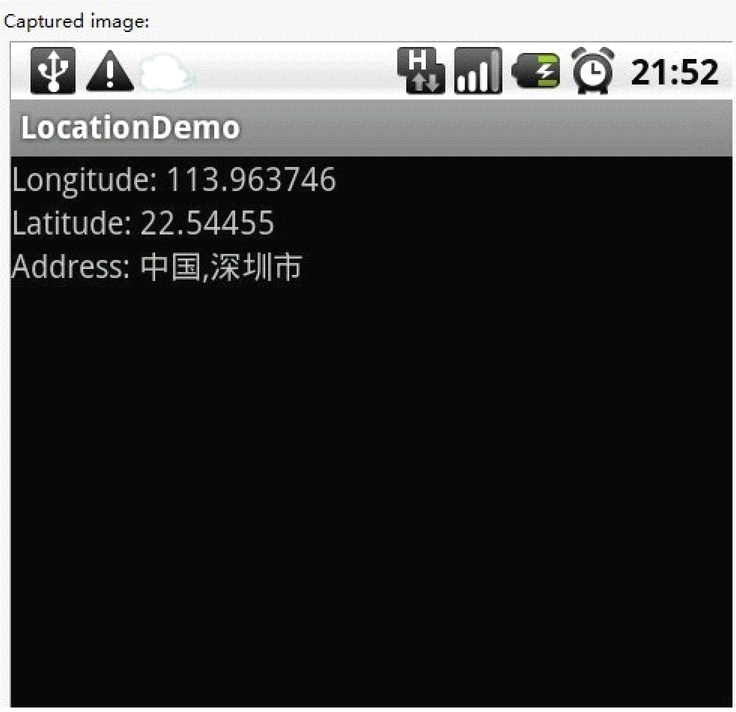
view plaincopy to clipboardprint?

```
<?xml version=″1.0″ encoding=″utf-8″?>
<manifest xmlns:android=″http://schemas.android.com/apk/res/android″
          package=″com.android.tutor″
          android:versionCode=″1″
          android:versionName=″1.0″>
                               <application      android:icon=″@drawable/icon″
android:label=″@string/app_name″>
              <activity android:name=″.LocationDemo″
                              android:label=″@string/app_name″>
                    <intent-filter>
                              <action android:name=″android.intent.action.MAIN″
/>
                                                                        <category
android:name=″android.intent.category.LAUNCHER″ />
                    </intent-filter>
              </activity>
              <uses-library android:name=″com.google.android.maps″ />
      </application>
      <uses-sdk android:minSdkVersion=″7″ />
                                                                 <uses-permission
android:name=″android.permission.ACCESS_FINE_LOCATION″/>
</manifest>
```

```
<?xml version=″1.0″ encoding=″utf-8″?>
<manifest xmlns:android=″http://schemas.android.com/apk/res/android″
          package=″com.android.tutor″
          android:versionCode=″1″
          android:versionName=″1.0″>
```

```
                                <application       android:icon="@drawable/icon"
android:label="@string/app_name">
              <activity android:name=".LocationDemo"
                              android:label="@string/app_name">
                    <intent-filter>
                         <action android:name="android.intent.action.MAIN" />
                                                                        <category
android:name="android.intent.category.LAUNCHER" />
                    </intent-filter>
              </activity>
         <uses-library android:name="com.google.android.maps" />
      </application>
      <uses-sdk android:minSdkVersion="7" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
</manifest>
```

第五步:运行上述工程，效果如下图如示:



Captured image:

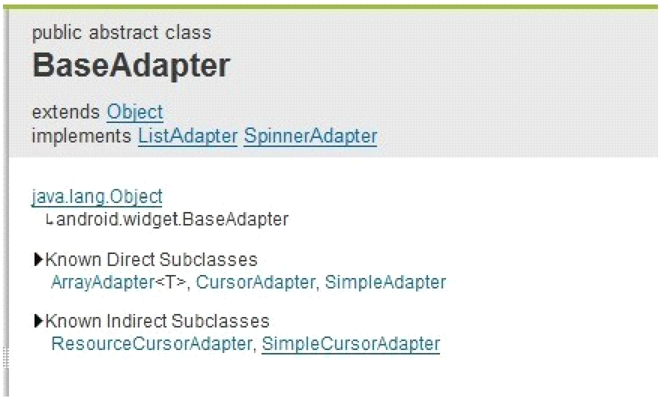
21:52

LocationDemo

Longitude: 113.963746
Latitude: 22.54455
Address: 中国,深圳市

Android 中万能的 BaseAdapter(Spinner,ListView,GridView)的使用

大家好!今天给大家讲解一下 BaseAdapter(基础适配器）的用法，适配器的作用主要是用来给诸如 (Spinner,ListView,GridView)来填充数据的。而(Spinner,ListView,GridView)都有自己的适配器(记 起来麻烦)。但是 BaseAdapter（一招鲜）对他们来说却是通用的，为什么这么说呢，首先我们看一下 API 文档:



public abstract class
BaseAdapter

extends Object
implements ListAdapter SpinnerAdapter

java.lang.Object
 ↳android.widget.BaseAdapter

▶Known Direct Subclasses
ArrayAdapter<T>, CursorAdapter, SimpleAdapter

▶Known Indirect Subclasses
ResourceCursorAdapter, SimpleCursorAdapter

我们看一下 BaseAdapter 已经实现了 ListAdapter 和 SpinnerAdapter 的接口，而 GridView 的适配器是实现了 ListAdapter 接口，只不过是二维的。所以说 BaseAdapter 对他们三者来说是通用的。

下面我来说一下 BaseAdapter 的主要用法.就是我们定义一个类(如：MyAdapter)而这个类继承 BaseAdapter.因为它是 implements 了 ListAdapter 和 SpinnerAdapter 的接口，所以要实现里面的方法，代码如下(未作任何改动的）：

view plaincopy to clipboardprint?

```
private class MyAdapter extends BaseAdapter{
      @Override
      public int getCount() {
              // TODO Auto-generated method stub
              return 0;
      }
      @Override
      public Object getItem(int arg0) {
              // TODO Auto-generated method stub
              return null;
      }
      @Override
      public long getItemId(int position) {
              // TODO Auto-generated method stub
              return 0;
      }
      @Override
      public View getView(int position, View convertView, ViewGroup parent) {

              // TODO Auto-generated method stub
              return null;
      }

     }
private class MyAdapter extends BaseAdapter{
    @Override
    public int getCount() {
     // TODO Auto-generated method stub
     return 0;
    }
    @Override
    public Object getItem(int arg0) {
     // TODO Auto-generated method stub
     return null;
    }
    @Override
    public long getItemId(int position) {
     // TODO Auto-generated method stub
     return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
     // TODO Auto-generated method stub
     return null;
    }


      }
```

为了便于大家理解，老规矩写一个简单的 Demo,大家按我的步骤来就 OK 了.

第一步:新建一个 Android 工程命名为 BaseAdapterDemo.

第二步:修改 main.xml 代码如下:

view plaincopy to clipboardprint?

```
<?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"
      >
      <TextView
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:text="Welcome to Mr Wei's Blog"
      />
      <Spinner
```

```
              android:id="@+id/spinner"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
      />
          <ListView
              android:id="@+id/listview"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
      />
          <GridView
              android:id="@+id/gridview"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
      />
</LinearLayout>
<?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"
      >
<TextView
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
         android:text="Welcome to Mr Wei's Blog"
      />
      <Spinner
         android:id="@+id/spinner"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
      />
          <ListView
         android:id="@+id/listview"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
      />
          <GridView
         android:id="@+id/gridview"
         android:layout_width="fill_parent"
         android:layout_height="wrap_content"
      />
</LinearLayout>
```

第三步:修该 BaseAdapterDemo.java 代码如下:

view plaincopy to clipboardprint?

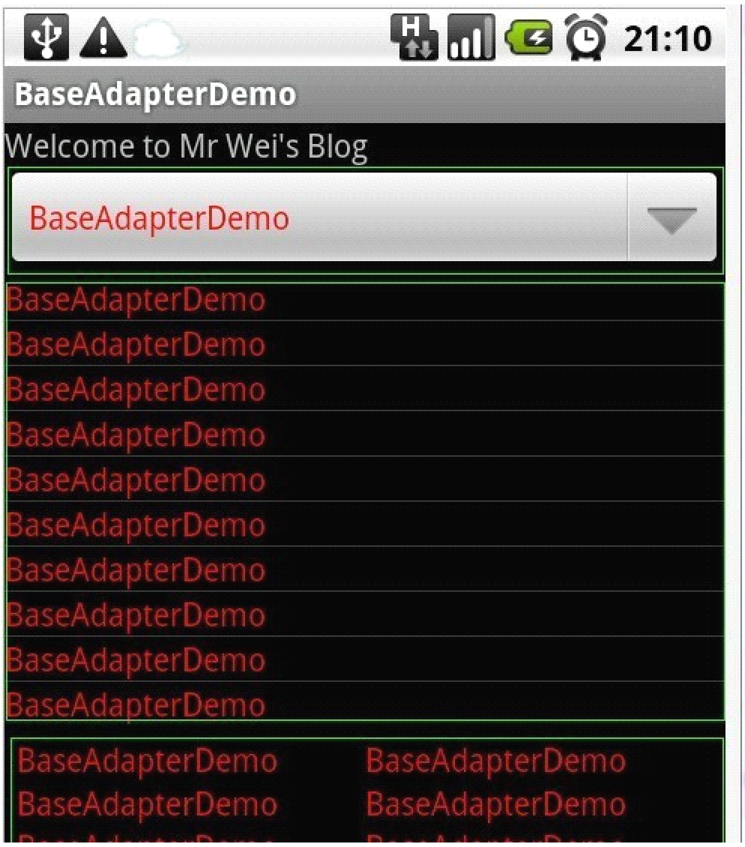
```
package com.tutor.baseadapter;
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
public class BaseAdapterDemo extends Activity {

      private Spinner mSpinner;
      private ListView mListView;
      private GridView mGridView;
      private MyAdapter mMyAdapter;
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
              setupViews();
      }

      public void setupViews(){
              mMyAdapter = new MyAdapter();
              mSpinner = (Spinner)findViewById(R.id.spinner);
              mSpinner.setAdapter(mMyAdapter);
              mListView = (ListView)findViewById(R.id.listview);
              mListView.setAdapter(mMyAdapter);
              mGridView = (GridView)findViewById(R.id.gridview);
              mGridView.setAdapter(mMyAdapter);
              mGridView.setNumColumns(2);

      }

      //定义自己的适配器,注意 getCount 和 getView 方法
      private class MyAdapter extends BaseAdapter{
              @Override
              public int getCount() {
                    // 这里我就返回 10 了，也就是一共有 10 项数据项
                    return 10;
              }
```

```
            @Override
            public Object getItem(int arg0) {
                  return arg0;
            }
            @Override
            public long getItemId(int position) {
                  return position;
            }
            @Override
            public View getView(int position, View convertView, ViewGroup parent)
{
                  // position 就是位置从 0 开始，convertView 是 Spinner,ListView
中每一项要显示的 view
                  //通常 return 的 view 也就是 convertView
                  //parent 就是父窗体了，也就是 Spinner,ListView,GridView
了.
                  TextView mTextView = new
TextView(getApplicationContext());
                  mTextView.setText("BaseAdapterDemo");
                  mTextView.setTextColor(Color.RED);
                  return mTextView;
            }

      }
}
package com.tutor.baseadapter;
import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ListView;
import android.widget.Spinner;
import android.widget.TextView;
public class BaseAdapterDemo extends Activity {

private Spinner mSpinner;
private ListView mListView;
private GridView mGridView;
private MyAdapter mMyAdapter;
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
              setupViews();
      }

      public void setupViews(){
         mMyAdapter = new MyAdapter();
         mSpinner = (Spinner)findViewById(R.id.spinner);
         mSpinner.setAdapter(mMyAdapter);
         mListView = (ListView)findViewById(R.id.listview);
         mListView.setAdapter(mMyAdapter);
         mGridView = (GridView)findViewById(R.id.gridview);
         mGridView.setAdapter(mMyAdapter);
         mGridView.setNumColumns(2);
      }

      //定义自己的适配器,注意 getCount 和 getView 方法
      private class MyAdapter extends BaseAdapter{
    @Override
    public int getCount() {
     // 这里我就返回 10 了，也就是一共有 10 项数据项
     return 10;
    }
    @Override
    public Object getItem(int arg0) {
     return arg0;
    }
    @Override
    public long getItemId(int position) {
     return position;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
     // position 就是位置从 0 开始，convertView 是 Spinner,ListView 中每一项要显示
的 view
     //通常 return 的 view 也就是 convertView
     //parent 就是父窗体了，也就是 Spinner,ListView,GridView 了.
     TextView mTextView = new TextView(getApplicationContext());
     mTextView.setText("BaseAdapterDemo");
     mTextView.setTextColor(Color.RED);
     return mTextView;
    }
```

```
        }
}
```

第四步:运行程序效果图如下:

效果图一:



21:10

BaseAdapterDemo

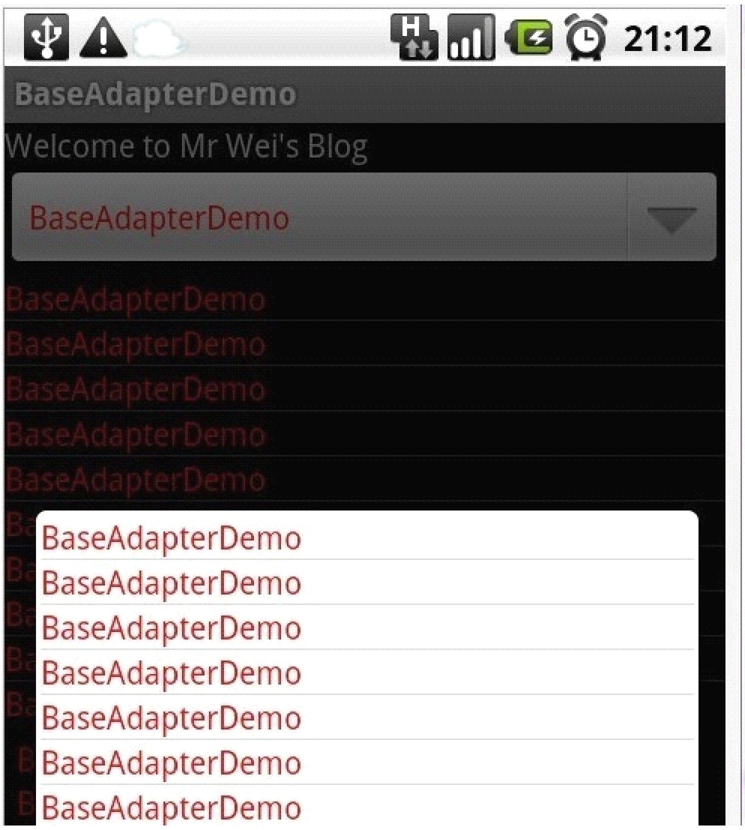
Welcome to Mr Wei's Blog

BaseAdapterDemo

BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo
BaseAdapterDemo

BaseAdapterDemo BaseAdapterDemo
BaseAdapterDemo BaseAdapterDemo

效果图二：



等等，平时我在这里就和大家告别了，今天还没完呵呵，因为下面是我们的重点了，我们平常看的应用列表什么的，不是单单的一个 TextView 就可以了事 的，所以我们可以在 Layout 里事先 定义好布局。这里我新建了一个名叫 baseadapter_provider.xml 文件，代码如下:

view plaincopy to clipboardprint?

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:orientation="horizontal"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      >
      <ImageView
              android:id="@+id/imageView"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:src="@drawable/icon"
      />
      <TextView
              android:id="@+id/textview"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="BaseAdapter"
      />
</LinearLayout>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
      android:orientation="horizontal"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent"
      >
      <ImageView
         android:id="@+id/imageView"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:src="@drawable/icon"
      />
<TextView
    android:id="@+id/textview"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="BaseAdapter"
    />
</LinearLayout>
```

将 getView()方法修改如下:

view plaincopy to clipboardprint?

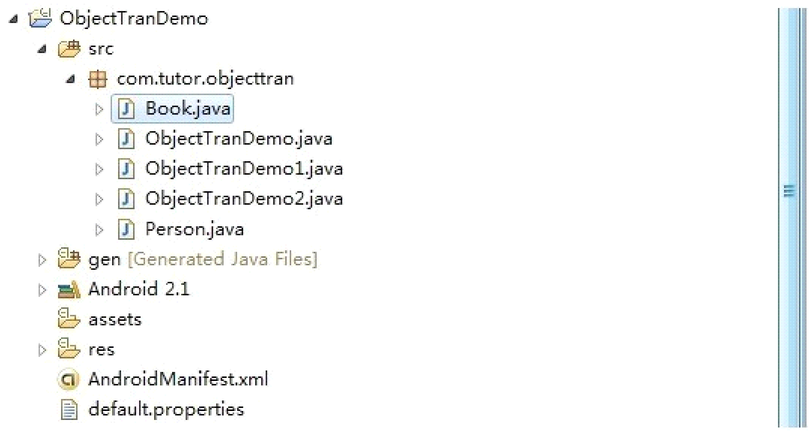
```
    @Override
         public View getView(int position, View convertView, ViewGroup parent)
{
             // position 就是位置从 0 开始，convertView 是 Spinner,ListView
中每一项要显示的 view
             //通常 return 的 view 也就是 convertView
               //parent 就是父窗体了，也就是 Spinner,ListView,GridView
了.
//                                          TextView    mTextView    =    new
TextView(getApplicationContext());
//            mTextView.setText("BaseAdapterDemo");
//            mTextView.setTextColor(Color.RED);
//            return mTextView;

               //LayoutInflater 不会的参照我的 Android 高手进阶教程
(五)
                                                      convertView    =
LayoutInflater.from(getApplicationContext()).inflate
             (R.layout.baseadapter_provider,null);

                                                 TextView    mTextView    =
(TextView)convertView.findViewById(R.id.textview);
                    mTextView.setText("BaseAdapterDemo" + position);
                    mTextView.setTextColor(Color.RED);
                    return convertView;
              }
@Override
    public View getView(int position, View convertView, ViewGroup parent) {
     // position 就是位置从 0 开始，convertView 是 Spinner,ListView 中每一项要显示
的 view
     //通常 return 的 view 也就是 convertView
     //parent 就是父窗体了，也就是 Spinner,ListView,GridView 了.
//     TextView mTextView = new TextView(getApplicationContext());
//     mTextView.setText("BaseAdapterDemo");
//     mTextView.setTextColor(Color.RED);
//     return mTextView;

     //LayoutInflater 不会的参照我的 Android 高手进阶教程(五)
     convertView = LayoutInflater.from(getApplicationContext()).inflate
     (R.layout.baseadapter_provider,null);

     TextView mTextView = (TextView)convertView.findViewById(R.id.textview);
     mTextView.setText("BaseAdapterDemo" + position);
     mTextView.setTextColor(Color.RED);
     return convertView;
    }
```

再次运行看一下效果图如下:

Android 中 Intent 传递对象的两种方法(Serializable,Parcelable)

大家好，好久不见，今天要给大家讲一下 Android 中 Intent 中如何传递对象,就我目前所知道的有两种方法，一种是 Bundle.putSerializable(Key,Object);另一种是 Bundle.putParcelable(Key, Object);当然这些 Object 是有一定的条件的，前者是实现了 Serializable 接口，而后者是实现了 Parcelable 接口，为了让大 家更容易理解我还是照常写了一个简单的 Demo,大家就一步一步跟我来吧！

第一步:新建一个 Android 工程命名为 ObjectTranDemo(类比较多哦!)目录结构如下图:



```
ObjectTranDemo
  src
    com.tutor.objecttran
      Book.java
      ObjectTranDemo.java
      ObjectTranDemo1.java
      ObjectTranDemo2.java
      Person.java
  gen [Generated Java Files]
  Android 2.1
  assets
  res
  AndroidManifest.xml
  default.properties
```

第二步:修改 main.xml 布局文件(这里我增加了两个按钮)代码如下 view plaincopy to clipboardprint?

```
<?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″
      >
```

```
<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Welcome to Mr wei's blog."
        />
<Button
        android:id="@+id/button1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Serializable"
/>
<Button
        android:id="@+id/button2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Parcelable"
/>
</LinearLayout>
<?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"
        >
<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Welcome to Mr wei's blog."
        />
```

```
<Button
android:id=″@+id/button1″
android:layout_width=″fill_parent″
android:layout_height=″wrap_content″
android:text=″Serializable″
/>
<Button
android:id=″@+id/button2″
android:layout_width=″fill_parent″
android:layout_height=″wrap_content″
android:text=″Parcelable″
/>
</LinearLayout>
```

第三步:新建两个类一个是Person.java实现Serializable接口，另一个Book.java实现Parcelable接口，代码分别如下:

Person.java:

view plaincopy to clipboardprint?

```
package com.tutor.objecttran;
import java.io.Serializable;
public class Person implements Serializable {
      private static final long serialVersionUID = -7060210544600464481L;
      private String name;
      private int age;
      public String getName() {
              return name;
      }
      public void setName(String name) {
              this.name = name;
      }
      public int getAge() {
              return age;
      }
      public void setAge(int age) {
              this.age = age;
      }

}
package com.tutor.objecttran;
import java.io.Serializable;
public class Person implements Serializable {
private static final long serialVersionUID = -7060210544600464481L;
private String name;
private int age;
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public int getAge() {
    return age;
}
public void setAge(int age) {
    this.age = age;
}

}
```

Book.java:

view plaincopy to clipboardprint?

```
package com.tutor.objecttran;
import android.os.Parcel;
import android.os.Parcelable;
public class Book implements Parcelable {
      private String bookName;
      private String author;
      private int publishTime;

      public String getBookName() {
              return bookName;
      }
      public void setBookName(String bookName) {
              this.bookName = bookName;
      }
      public String getAuthor() {
              return author;
      }
      public void setAuthor(String author) {
              this.author = author;
      }
      public int getPublishTime() {
              return publishTime;
      }
      public void setPublishTime(int publishTime) {
              this.publishTime = publishTime;
      }
      public static final Parcelable.Creator<Book> CREATOR = new Creator<Book>()
{
              public Book createFromParcel(Parcel source) {
                    Book mBook = new Book();
                    mBook.bookName = source.readString();
                    mBook.author = source.readString();
                    mBook.publishTime = source.readInt();
                    return mBook;
              }
              public Book[] newArray(int size) {
                    return new Book[size];
              }
      };

      public int describeContents() {
              return 0;
      }
      public void writeToParcel(Parcel parcel, int flags) {
              parcel.writeString(bookName);
              parcel.writeString(author);
              parcel.writeInt(publishTime);
      }
}
package com.tutor.objecttran;
import android.os.Parcel;
import android.os.Parcelable;
public class Book implements Parcelable {
private String bookName;
private String author;
private int publishTime;

public String getBookName() {
    return bookName;
}
public void setBookName(String bookName) {
    this.bookName = bookName;
}
public String getAuthor() {
    return author;
}
public void setAuthor(String author) {
    this.author = author;
}
public int getPublishTime() {
    return publishTime;
}
public void setPublishTime(int publishTime) {
    this.publishTime = publishTime;
}

public static final Parcelable.Creator<Book> CREATOR = new Creator<Book>() {
    public Book createFromParcel(Parcel source) {
     Book mBook = new Book();
     mBook.bookName = source.readString();
     mBook.author = source.readString();
     mBook.publishTime = source.readInt();
     return mBook;
    }
    public Book[] newArray(int size) {
     return new Book[size];
    }
};

public int describeContents() {
    return 0;
}
public void writeToParcel(Parcel parcel, int flags) {
    parcel.writeString(bookName);
    parcel.writeString(author);
    parcel.writeInt(publishTime);
}
}
```

第四步:修改 ObjectTranDemo.java,并且新建两个 Activity,一个是 ObjectTranDemo1.java,别一个是 ObjectTranDemo2.java.分别用来显示 Person 对像数据，和 Book 对象数据:,代码分别如下：

ObjectTranDemo.java:

view plaincopy to clipboardprint?

```
package com.tutor.objecttran;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class ObjectTranDemo extends Activity implements OnClickListener {
```

```
private Button sButton,pButton;
public    final static String SER_KEY = "com.tutor.objecttran.ser";
public    final static String PAR_KEY = "com.tutor.objecttran.par";
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        setupViews();

}

//我的一贯作风呵呵
public void setupViews(){
        sButton = (Button)findViewById(R.id.button1);
        pButton = (Button)findViewById(R.id.button2);
        sButton.setOnClickListener(this);
        pButton.setOnClickListener(this);
}
//Serializeable 传递对象的方法
public void SerializeMethod(){
        Person mPerson = new Person();
        mPerson.setName("frankie");
        mPerson.setAge(25);
        Intent mIntent = new Intent(this,ObjectTranDemo1.class);
        Bundle mBundle = new Bundle();
        mBundle.putSerializable(SER_KEY,mPerson);
        mIntent.putExtras(mBundle);

        startActivity(mIntent);
}
```

```
      //Pacelable 传递对象方法
      public void PacelableMethod(){
              Book mBook = new Book();
              mBook.setBookName("Android Tutor");
              mBook.setAuthor("Frankie");
              mBook.setPublishTime(2010);
              Intent mIntent = new Intent(this,ObjectTranDemo2.class);
              Bundle mBundle = new Bundle();
              mBundle.putParcelable(PAR_KEY, mBook);
              mIntent.putExtras(mBundle);

              startActivity(mIntent);
      }
      //铵钮点击事件响应
      public void onClick(View v) {
              if(v == sButton){
                    SerializeMethod();
              }else{
                    PacelableMethod();
              }
      }
}
package com.tutor.objecttran;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```

```
public class ObjectTranDemo extends Activity implements OnClickListener {


private Button sButton,pButton;
public    final static String SER_KEY = "com.tutor.objecttran.ser";
public    final static String PAR_KEY = "com.tutor.objecttran.par";
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.main);
              setupViews();

      }

      //我的一贯作风呵呵
      public void setupViews(){
         sButton = (Button)findViewById(R.id.button1);
         pButton = (Button)findViewById(R.id.button2);
         sButton.setOnClickListener(this);
         pButton.setOnClickListener(this);
      }
      //Serializeable 传递对象的方法
      public void SerializeMethod(){
         Person mPerson = new Person();
         mPerson.setName("frankie");
         mPerson.setAge(25);
         Intent mIntent = new Intent(this,ObjectTranDemo1.class);
         Bundle mBundle = new Bundle();
         mBundle.putSerializable(SER_KEY,mPerson);
         mIntent.putExtras(mBundle);
         startActivity(mIntent);
      }
      //Pacelable 传递对象方法
      public void PacelableMethod(){
         Book mBook = new Book();
         mBook.setBookName("Android Tutor");
         mBook.setAuthor("Frankie");
         mBook.setPublishTime(2010);
         Intent mIntent = new Intent(this,ObjectTranDemo2.class);
         Bundle mBundle = new Bundle();
         mBundle.putParcelable(PAR_KEY, mBook);
         mIntent.putExtras(mBundle);

         startActivity(mIntent);
      }
      //铵钮点击事件响应
public void onClick(View v) {
    if(v == sButton){
     SerializeMethod();
    }else{
     PacelableMethod();
    }
}
}
```

ObjectTranDemo1.java:

view plaincopy to clipboardprint?

```
package com.tutor.objecttran;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class ObjectTranDemo1 extends Activity {
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);

              TextView mTextView = new TextView(this);
              Person mPerson = 
(Person)getIntent().getSerializableExtra(ObjectTranDemo.SER_KEY);
              mTextView.setText("You name is: " + mPerson.getName() + "\n"+

                          "You age is: " + mPerson.getAge());

              setContentView(mTextView);
      }
}
package com.tutor.objecttran;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class ObjectTranDemo1 extends Activity {
      @Override
      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);

              TextView mTextView = new TextView(this);
              Person mPerson = 
(Person)getIntent().getSerializableExtra(ObjectTranDemo.SER_KEY);
              mTextView.setText("You name is: " + mPerson.getName() + "\n"+
                "You age is: " + mPerson.getAge());

              setContentView(mTextView);
      }
}
```

ObjectTranDemo2.java:

view plaincopy to clipboardprint?

```
package com.tutor.objecttran;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class ObjectTranDemo2 extends Activity {

      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              TextView mTextView = new TextView(this);
              Book mBook =
(Book)getIntent().getParcelableExtra(ObjectTranDemo.PAR_KEY);
              mTextView.setText("Book name is: " + mBook.getBookName()+"\n"+

                                            "Author is: " + mBook.getAuthor() +
"\n" +
                                            "PublishTime is: " +
mBook.getPublishTime());
              setContentView(mTextView);
      }
}
package com.tutor.objecttran;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class ObjectTranDemo2 extends Activity {

      public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              TextView mTextView = new TextView(this);
              Book mBook =
(Book)getIntent().getParcelableExtra(ObjectTranDemo.PAR_KEY);
              mTextView.setText("Book name is: " + mBook.getBookName()+"\n"+
                        "Author is: " + mBook.getAuthor() + "\n" +
                        "PublishTime is: " + mBook.getPublishTime());
              setContentView(mTextView);
      }
}
```

第五步:比较重要的一步啦，修改 AndroidManifest.xml 文件（将两个新增的 Activity，ObjectTranDemo1,ObjectTranDemo2)申明一下代码如下(第 14，15 行):

view plaincopy to clipboardprint?

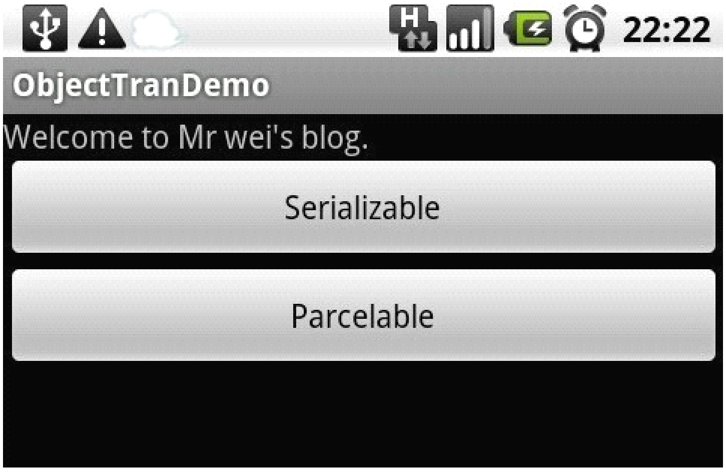
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.tutor.objecttran"
          android:versionCode="1"
          android:versionName="1.0">
      <application android:icon="@drawable/icon"
android:label="@string/app_name">
              <activity android:name=".ObjectTranDemo"
                              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=".ObjectTranDemo1"></activity>
              <activity android:name=".ObjectTranDemo2"></activity>
      </application>
      <uses-sdk android:minSdkVersion="7" />
</manifest>
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="com.tutor.objecttran"
          android:versionCode="1"
          android:versionName="1.0">
      <application android:icon="@drawable/icon"
android:label="@string/app_name">
              <activity android:name=".ObjectTranDemo"
                              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=".ObjectTranDemo1"></activity>
    <activity android:name=".ObjectTranDemo2"></activity>
      </application>
      <uses-sdk android:minSdkVersion="7" />
```

```
</manifest>
```

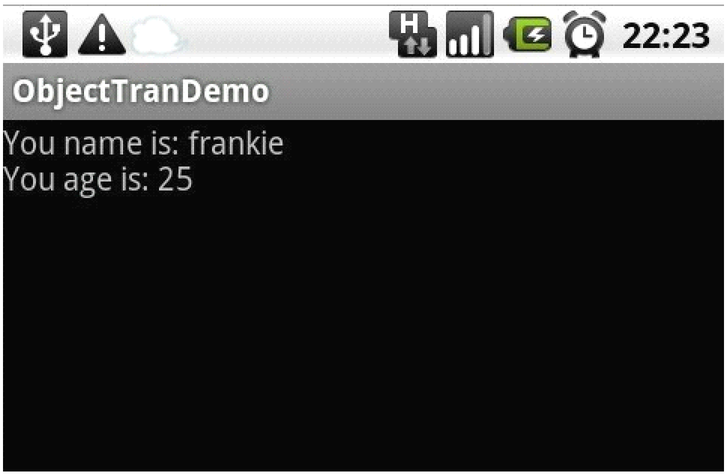
第六步:运行上述工程查看效果图啦:

效果 1：首界面:



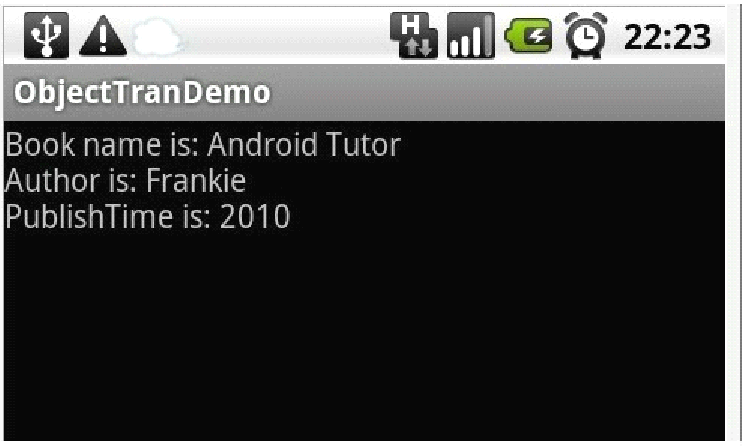
Welcome to Mr wei's blog.

效果 2:点击 Serializable 按钮



You name is: frankie
You age is: 25

效果 3:点击 Parcelable 按钮:



```
Book name is: Android Tutor
Author is: Frankie
PublishTime is: 2010
```