[Урок 49 SimpleAdapter Методы SetViewText и SetViewImage](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/108-urok-49-simpleadapter-metody-setviewtext-i-setviewimage.html" \o "Урок 49. SimpleAdapter. Методы SetViewText и SetViewImage)

19 января 2012

В этом уроке:

- используем методы SetViewText и SetViewImage

Мы уже знаем, что **SimpleAdapter**умеет вставлять текст в TextView элементы и изображения в ImageView. Он использует для этого методы **SetViewText**и **SetViewImage**. Мы можем создать **свой адаптер** на основе SimpleAdapter и реализовать эти методы под наши цели.

Эти методы предоставляют нам **View**и **данные**, а значит мы можем менять View в зависимости от данных. В качестве примера, сделаем список, отражающий динамику некоего показателя в разрезе дней. Если динамика положительная – будем разукрашивать элементы в зеленый цвет, если отрицательная – в красный.

Создадим проект:

**Project name**: P0491\_SimpleAdapterCustom1  
**Build Target**: Android 2.3.3  
**Application name**: SimpleAdapterCustom1  
**Package name**: ru.startandroid.develop.p0491simpleadaptercustom1  
**Create Activity**: MainActivity

Рисуем экран **main.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="fill\_parent"      android:layout\_height="fill\_parent"      android:orientation="vertical">      <ListView          android:id="@+id/lvSimple"          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content">      </ListView>  </LinearLayout> |

На экране только список.

И layout для пункта списка **item.xml**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout      xmlns:android="http://schemas.android.com/apk/res/android"      android:layout\_width="match\_parent"      android:layout\_height="wrap\_content"      android:orientation="horizontal">      <LinearLayout          android:layout\_width="match\_parent"          android:layout\_height="wrap\_content"          android:layout\_marginBottom="5dp"          android:layout\_marginTop="5dp"          android:orientation="horizontal">          <ImageView              android:id="@+id/ivImg"              android:layout\_width="30dp"              android:layout\_height="22dp"              android:scaleType="fitCenter">          </ImageView>          <TextView              android:id="@+id/tvValue"              android:layout\_width="wrap\_content"              android:layout\_height="wrap\_content"              android:layout\_marginLeft="10dp"              android:text="TextView"              android:textSize="20sp">          </TextView>          <FrameLayout              android:layout\_width="match\_parent"              android:layout\_height="wrap\_content">              <TextView                  android:id="@+id/tvText"                  android:layout\_width="wrap\_content"                  android:layout\_height="wrap\_content"                  android:layout\_gravity="center\_vertical|center\_horizontal"                  android:text="TextView"                  android:textSize="20sp">              </TextView>          </FrameLayout>      </LinearLayout>  </LinearLayout> |

Компонент **ivImg**будет отображать стрелку вниз или вверх, **tvValue**– значение динамики, **tvText**– номер дня.

Код **MainActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97 | package ru.startandroid.develop.p0491simpleadaptercustom1;    import java.util.ArrayList;  import java.util.HashMap;  import java.util.List;  import java.util.Map;    import android.app.Activity;  import android.content.Context;  import android.graphics.Color;  import android.os.Bundle;  import android.widget.ImageView;  import android.widget.ListView;  import android.widget.SimpleAdapter;  import android.widget.TextView;    public class MainActivity extends Activity {      // имена атрибутов для Map    final String ATTRIBUTE\_NAME\_TEXT = "text";    final String ATTRIBUTE\_NAME\_VALUE = "value";    final String ATTRIBUTE\_NAME\_IMAGE = "image";      // картинки для отображения динамики    final int positive = android.R.drawable.arrow\_up\_float;    final int negative = android.R.drawable.arrow\_down\_float;      ListView lvSimple;      /\*\* Called when the activity is first created. \*/    public void onCreate(Bundle savedInstanceState) {      super.onCreate(savedInstanceState);      setContentView(R.layout.main);        // массив данных      int[] values = { 8, 4, -3, 2, -5, 0, 3, -6, 1, -1 };        // упаковываем данные в понятную для адаптера структуру      ArrayList<Map<String, Object>> data = new ArrayList<Map<String, Object>>(          values.length);      Map<String, Object> m;      int img = 0;      for (int i = 0; i < values.length; i++) {        m = new HashMap<String, Object>();        m.put(ATTRIBUTE\_NAME\_TEXT, "Day " + (i + 1));        m.put(ATTRIBUTE\_NAME\_VALUE, values[i]);        if (values[i] == 0) img = 0; else          img = (values[i] > 0) ? positive : negative;        m.put(ATTRIBUTE\_NAME\_IMAGE, img);        data.add(m);      }        // массив имен атрибутов, из которых будут читаться данные      String[] from = { ATTRIBUTE\_NAME\_TEXT, ATTRIBUTE\_NAME\_VALUE,          ATTRIBUTE\_NAME\_IMAGE };      // массив ID View-компонентов, в которые будут вставлять данные      int[] to = { R.id.tvText, R.id.tvValue, R.id.ivImg };        // создаем адаптер      MySimpleAdapter sAdapter = new MySimpleAdapter(this, data,          R.layout.item, from, to);        // определяем список и присваиваем ему адаптер      lvSimple = (ListView) findViewById(R.id.lvSimple);      lvSimple.setAdapter(sAdapter);    }      class MySimpleAdapter extends SimpleAdapter {        public MySimpleAdapter(Context context,          List<? extends Map<String, ?>> data, int resource,          String[] from, int[] to) {        super(context, data, resource, from, to);      }        @Override      public void setViewText(TextView v, String text) {        // метод супер-класса, который вставляет текст        super.setViewText(v, text);        // если нужный нам TextView, то разрисовываем        if (v.getId() == R.id.tvValue) {          int i = Integer.parseInt(text);          if (i < 0) v.setTextColor(Color.RED); else            if (i > 0) v.setTextColor(Color.GREEN);        }      }        @Override      public void setViewImage(ImageView v, int value) {        // метод супер-класса        super.setViewImage(v, value);        // разрисовываем ImageView        if (value == negative) v.setBackgroundColor(Color.RED); else          if (value == positive) v.setBackgroundColor(Color.GREEN);      }    }  } |

Код создания адаптера обычен. Придумываем данные **values**и упаковываем их в коллекцию **Map**-объектов. Каждый Map будет состоять из трех атрибутов

ATTRIBUTE\_NAME\_TEXT - текст с номером дня  
ATTRIBUTE\_NAME\_VALUE - значение динамики  
ATTRIBUTE\_NAME\_IMAGE - id картинки для отображения, в зависимости от значения (положительное или отрицательное)

Затем заполняем массивы сопоставления данных (from) и View (to), создаем адаптер, используя свой класс **MySimpleAdapter**, и настраиваем список.

Смотрим реализацию **MySimpleAdapter**. Конструктор вызывает конструктор супер-класса, тут ничего не меняем. Менять будем методы:

**setViewText**

Сначала выполняем метод супер-класса, который вставляет данные. А далее смотрим, если View, это тот TextView, который будет отображать значения, то меняем цвет текста на красный или зеленый в зависимости от значения, которое он будет отображать.

**setViewImage**

Выполняем метод супер-класса, чтобы ImageView получил изображение, а дальше меняем его фон в зависимости от значения. Картинка, которую мы вставляем , с альфа-слоем, поэтому фон будет виден. Проверку по id ImageView не выполняем, т.к. ImageView у нас только один.

Все сохраним и запускаем.



Отобразились картинка, значение и день. Фон картинки и цвет текста значения меняются в зависимости от значения – положительное или отрицательное.

Пример, конечно, не особо интересный, но надо было рассмотреть работу этих методов. Ничего лучше в голову не пришло )

На следующем уроке:

- используем свой SimpleAdapter.ViewBinder

Похожие статьи

* [Урок 118. Виджеты. Конфигурационный экран. Обновление](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/198-urok-118-vidzhety-konfiguratsionnyj-ekran-obnovlenie.html)
* [Урок 52. SimpleCursorAdapter, пример использования](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/111-urok-52-simplecursoradapter-primer-ispolzovanija.html)
* [Урок 51. SimpleAdapter, добавление и удаление записей](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/110-urok-51-simpleadapter-dobavlenie-i-udalenie-zapisej.html)
* [Урок 50. SimpleAdapter. Используем ViewBinder](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/109-urok-50-simpleadapter-ispolzuem-viewbinder.html)
* [Урок 48. Используем SimpleAdapter.](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/107-urok-48-ispolzuem-simpleadapter.html)
* [Урок 47. Обзор адаптеров](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/96-urok-47-obzor-adapterov.html)

Последние статьи

* [Урок 176. OpenGL. Индексы, текстуры для куба.](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/408-urok-176-opengl-indeksy-tekstury-dlja-kuba.html)
* [Урок 175. OpenGL. Текстуры.](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/406-urok-175-opengl-tekstury.html)
* [Урок 174. OpenGL. Модель](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/405-urok-174-opengl-model.html)
* [Урок 173. OpenGL. Камера](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/403-urok-173-opengl-kamera.html)
* [Урок 172. OpenGL. Perspective. Frustum. Ortho.](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/401-urok-172-perspective-frustum-ortho.html)