# **[Android 设计模式之MVC模式](http://www.cnblogs.com/liqw/p/4175325.html)**

说到Android设计模式的MVC模式，估计很多人都是比较熟悉了，这里深入了解一下MVC到底是怎么回事，以ListView为例子讲解。

**一、深入理解MVC概念**

MVC即Model-View-Controller。M：逻辑模型，V：视图模型，C：控制器。

MVC模式下，系统框架的类库被划分为3种：模型（Model）、视图（View）、控制器（Controller）。模型对象负责建立数据结构和相应的行为操作处理。视图对象负责在屏幕上渲染出相应的图形信息展示给用户看。控制器对象负责截获用户的按键和屏幕触摸等事件，协调Model对象和View对象。

用户与视图交互，视图接收并反馈用户的动作；视图把用户的请求传给相应的控制器，由控制器决定调用哪个模型，然后由模型调用相应的业务逻辑对用户请求进行加工处理，如果需要返回数据，模型会把相应的数据返回给控制器，由控制器调用相应的视图，最终由视图格式化和渲染返回的数据，对于返回的数据完全可以增加用户体验效果展现给用户。

一个模型可以有多个视图，一个视图可以有多个控制器，一个控制器也可以有多个模型。

（1）、模型（Model）

Model是一个应用系统的核心部分，代表了该系统实际要实现的所有功能处理。比如：在视频播放器中，模型代表一个视频数据库及播放视频的程序函数代码；在拍照应用中，模型代表一个照片数据库，及看图片时的程序函数代码。在一个电话应用中，Model代表一个电话号码簿，以及拨打电话和发送短信的程序函数代码。

Model在values目录下通过xml文件格式生成，也可以通过硬编码的方式直接Java代码生成。View和Model是通过桥梁Adapter来连接起来。

（2）、视图（View）

View是软件应用传送给用户的一个反馈结果。它代表软件应用中的图形展示、声音播放、触觉反馈等职责。视图的根节点是应用程序的自身窗口。比如，视频播放器中可能包含当前播放的画面，这个画面就是一个视图。另一个视图组件可能是该视频的文字标题。再一个就是一些播放按键，比如：Stop、Start、Pause等按钮。

View在layout目录下通过xml文件格式生成，用findViewById()获取；也可以通过硬编码的方式直接Java代码生成。

（3）、控制器（Controller）

Controller在软件应用负责对外部事件的响应，包括：键盘敲击、屏幕触摸、电话呼入等。Controller实现了一个事件队列，每一个外部事件均在事件队列中被唯一标识。框架依次将事件从队列中移出并派发出去。

**二、MVC典型例子的实现ListView**

Android中最典型MVC是ListView，要显示的数据是Model，界面中的ListView是View，控制数据怎样在ListView中显示是Controller。

**（1）、通过硬编码的方式直接Java代码生成方式，这里直接在注释讲解**

public class ArrayAdapterActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

//界面中的ListView是View，这里通过硬编码的方式直接Java代码生成

ListView listView = new ListView(this);

// 控制数据怎样在ListView中显示是Controller

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_expandable\_list\_item\_1, getData());

//View和Model是通过桥梁Adapter来连接起来。

listView.setAdapter(adapter);

setContentView(listView);

// 点击事件，Controller负责

listView.setOnItemClickListener(new OnItemClickListener() {

@Override

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

// position是从0开始的,获取点击item的内容

Toast.makeText(ArrayAdapterActivity.this, getData().get(position), Toast.LENGTH\_SHORT).show();

}

});

}

// 要显示的数据Model，通过硬编码的方式直接Java代码生成

private List<String> getData() {

List<String> data = new ArrayList<String>();

data.add("a");

data.add("b");

data.add("c");

data.add("d");

return data;

}

}

**（2）、视图View和模型Model取资源文件方式**

先在res/layout文件夹下创建文件activity\_arrayadapter.xml，可以看出只包含一个ListView，即视图View

<?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/listview"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:drawSelectorOnTop="false"/>

</LinearLayout>

在res/values文件夹下的strings.xml添加一个字符数组，及模型Model

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

<resources>

<string-array name="good">

<item>a</item>

<item>b</item>

<item>c</item>

<item>d</item>

</string-array>

</resources

Activity代码，在注释中讲解MVC模型使用\

public class ArrayAdapterActivity2 extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_arrayadapter);

//界面中的ListView是View，View在layout目录下通过xml文件格式生成，用findViewById()获取

ListView listView = (ListView) findViewById(R.id.listview);

// 控制数据怎样在ListView中显示是Controller

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_expandable\_list\_item\_1, getData());

//View和Model是通过桥梁Adapter来连接起来。

listView.setAdapter(adapter);

// 点击事件，Controller负责

listView.setOnItemClickListener(new OnItemClickListener() {

@Override

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

// position是从0开始的,获取点击item的内容

Toast.makeText(ArrayAdapterActivity2.this, getData().get(position), Toast.LENGTH\_SHORT).show();

}

});

}

// 要显示的数据Model，Model在values目录下通过xml文件格式生成

private List<String> getData() {

List<String> data = new ArrayList<String>();

Resources res =getResources();

// 取xml文件格式的字符数组

String[] good=res.getStringArray(R.array.good);

for(int i=0;i<good.length;i++){

data.add(good[i]);

}

return data;

}

}

Android的MVC模式要在项目中慢慢理解，这样才能理解透彻并活学活用。

到这里基本介绍完了Android设计模式之MVC模型，若本文对您有帮助请点“推荐”支持，若有什么错误请指正留言，共同探讨。

转载请附上本文链接：[http://www.cnblogs.com/liqw/p/4175325.html](http://www.cnblogs.com/liqw/p/4175325.html" \t "http://www.cnblogs.com/liqw/p/_blank)