

2.6.4 DrawerLayout(官方侧滑菜单)的简单使用

分类 [Android 基础入门教程](#)

[Android 基础入门教程\(Q群号 : 153836263\)](#)

[1.0 Android基础入门教程](#)

[1.0.1 2015年最新Android基...](#)

[1.1 背景相关与系统架构分析](#)

[1.2 开发环境搭建](#)

[1.2.1 使用Eclipse + ADT + S...](#)

[1.2.2 使用Android Studio开...](#)

[1.3 SDK更新不了问题解决](#)

[1.4 Genymotion模拟器安装](#)

[1.5.1 Git使用教程之本地仓...](#)

[1.5.2 Git之使用GitHub搭建...](#)

[1.6 .9\(九妹\)图片怎么玩](#)

[1.7 界面原型设计](#)

[1.8 工程相关解析\(各种文件...](#)

[1.9 Android程序签名打包](#)

[1.11 反编译APK获取代码&...](#)

[2.1 View与ViewGroup的概念](#)

[2.2.1 LinearLayout\(线性布局\)](#)

[2.2.2 RelativeLayout\(相对布...](#)

[2.2.3 TableLayout\(表格布局\)](#)

[2.2.4 FrameLayout\(帧布局\)](#)

[2.2.5 GridLayout\(网格布局\)](#)

[2.2.6 AbsoluteLayout\(绝对...](#)

[2.3.1 TextView\(文本框\)详解](#)

[2.3.2 EditText\(输入框\)详解](#)

[2.3.3 Button\(按钮\)与ImageB...](#)

[2.3.4 ImageView\(图像视图\)](#)

[2.3.5.RadioButton\(单选按钮...](#)

[2.3.6 开关按钮ToggleButton...](#)

[2.3.7 ProgressBar\(进度条\)](#)

[2.3.8 SeekBar\(拖动条\)](#)

[2.3.9 RatingBar\(星级评分条\)](#)

[2.4.1 ScrollView\(滚动条\)](#)

本节引言：

本节给大家带来基础UI控件部分的最后一个控件：**DrawerLayout**，官方给我们提供的一个侧滑菜单 控件，和上一节的ViewPager一样，3.0以后引入，低版本使用它，需要v4兼容包，说到侧滑，相信 很多人都用过github上的SlidingMenu，不过好像有两个版本，一个是单独的，另一个需要依赖另一个开源项目：ActionBarSherlock；既然Google为我们提供了这个控件，为何不用咧，而且在 Material Design设计规范中，随处可见的很多侧滑菜单的动画效果，大都可以通过Toolbar + DrawerLayout来实现~，本节我们就来探究下这个DrawerLayout的一个基本用法~还有人喜欢把他 称为抽屉控件~官方文档：[DrawerLayout](#)

1.使用的注意事项

1.主内容视图一定要是DrawerLayout的第一个子视图

2.主内容视图宽度和高度需要match_parent

3.必须显示指定侧滑视图的android:layout_gravity属性

android:layout_gravity = "start"时，从左向右滑出菜单

android:layout_gravity = "end"时，从右向左滑出菜单 不推荐使用left和right!!!

侧滑视图的宽度以dp为单位，不建议超过320dp(为了总能看到一些主内容视图)

设置侧滑事件：

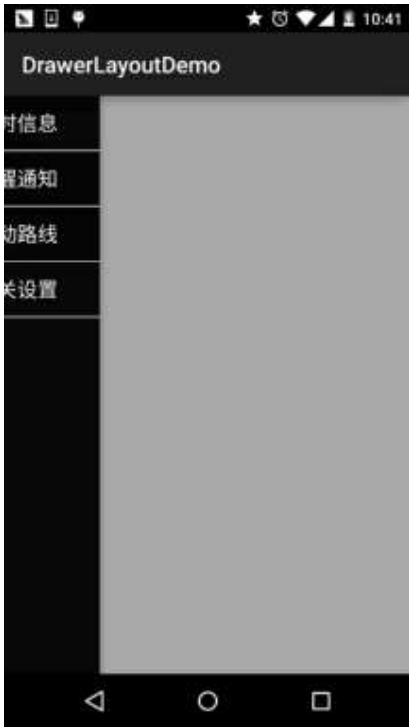
```
mDrawerLayout.setDrawerListener(DrawerLayout.DrawerListener);
```

要说一点：可以结合ActionBar使用当用户点击ActionBar上的应用图标，弹出侧滑菜单！这里就要通过**ActionBarDrawerToggle**，它是DrawerLayout.DrawerListener的具体实现类，我们可以重写ActionBarDrawerToggle的onDrawerOpened()和onDrawerClosed()以监听抽屉拉出 或隐藏事件！但是这里我们不讲，因为5.0后我们使用的是Toolbar！有兴趣的可以自行查阅相关 文档！

2.使用代码示例

示例1：单个侧滑菜单的实现

运行效果图：



实现关键代码：

首先是我们的主布局，注意：最外层要是DrawerLayout哦！！！！

activity_main.xml：

```
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <FrameLayout
        android:id="@+id/ly_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <ListView
        android:id="@+id/list_left_drawer"
        android:layout_width="180dp"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:background="#080808"
        android:choiceMode="singleChoice"
        android:divider="#FFFFFF"
        android:dividerHeight="1dp" />

</android.support.v4.widget.DrawerLayout>
```

接着ListView的布局代码和domain类：Item比较简单，就不给出了，直接上中间Fragment的布局以及代码吧！另外Adapter直接复用我们之前写的那个

2.4.2 Date & Time组件(上)

2.4.3 Date & Time组件(下)

2.4.4 Adapter基础讲解

2.4.5 ListView简单实用

2.4.6 BaseAdapter优化

2.4.7 ListView的焦点问题

2.4.8 ListView之checkbox错...

2.4.9 ListView的数据更新问题

2.5.0 构建一个可复用的自定...

2.5.1 ListView Item多布局的...

2.5.2 GridView(网格视图)的...

2.5.3 Spinner(列表选项框)...

2.5.4 AutoCompleteTextView...

2.5.5 ExpandableListView(...

2.5.6 ViewPager(翻转视图)...

2.5.7 Toast(吐司)的基本使用

2.5.8 Notification(状态栏通...

2.5.9 AlertDialog(对话框)详解

2.6.0 其他几种常用对话框基...

2.6.1 PopupWindow(悬浮框...

2.6.2 菜单(Menu)

2.6.3 ViewPager的简单使用

2.6.4 DrawerLayout(官方侧...

3.1.1 基于监听的事件处理机制

3.2 基于回调的事件处理机制

3.3 Handler消息传递机制浅析

3.4 TouchListener PK OnTo...

3.5 监听EditText的内容变化

3.6 响应系统设置的事件(Co...

3.7 AsyncTask异步任务

3.8 Gestures(手势)

4.1.1 Activity初学乍练

4.1.2 Activity初窥门径

4.1.3 Activity登堂入室

4.2.1 Service初涉

4.2.2 Service进阶

4.2.3 Service精通

4.3.1 BroadcastReceiver牛...

4.3.2 BroadcastReceiver庖...

4.4.1 ContentProvider初探

可复用的MyAdapter !

fg_content.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tv_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:textSize="25sp" />

</RelativeLayout>
```

ContentFragment.java :

```
/**
 * Created by Jay on 2015/10/8 0008.
 */
public class ContentFragment extends Fragment {

    private TextView tv_content;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_content, container, false);
        tv_content = (TextView) view.findViewById(R.id.tv_content);
        String text = getArguments().getString("text");
        tv_content.setText(text);
        return view;
    }
}
```

最后是我们的Activity类

MainActivity.java :

```
public class MainActivity extends AppCompatActivity implements
    AdapterView.OnItemClickListener{

    private DrawerLayout drawer_layout;
    private ListView list_left_drawer;
    private ArrayList<Item> menuLists;
    private MyAdapter<Item> myAdapter = null;
```

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概述

5.2.1 Fragment实例精讲—...

5.2.2 Fragment实例精讲—...

5.2.3 Fragment实例精讲—...

5.2.4 Fragment实例精讲—...

5.2.5 Fragment实例精讲—...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 (1)

7.3.3 Android 文件下载 (2)

7.4 Android 调用 Webservice

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScrip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

8.3.2 绘图类实战示例

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    drawer_layout = (DrawerLayout) findViewById(R.id.drawer
_layout);
    list_left_drawer = (ListView) findViewById(R.id.list_le
ft_drawer);

    menuLists = new ArrayList<Item>();
    menuLists.add(new Item(R.mipmap.iv_menu_realtime,"实时
信息"));
    menuLists.add(new Item(R.mipmap.iv_menu_alert,"提醒通知"
));
    menuLists.add(new Item(R.mipmap.iv_menu_trace,"活动路线"
));
    menuLists.add(new Item(R.mipmap.iv_menu_settings,"相关
设置"));
    myAdapter = new MyAdapter<Item>(menuLists,R.layout.item
_list) {
        @Override
        public void bindView(ViewHolder holder, Item obj) {
            holder.setImageResource(R.id.img_icon,obj.getIc
onId());
            holder.setText(R.id.txt_content, obj.getIconNam
e());
        }
    };
    list_left_drawer.setAdapter(myAdapter);
    list_left_drawer.setOnItemClickListener(this);
}

@Override
public void onItemClick(AdapterView<?> parent, View view, i
nt position, long id) {
    ContentFragment contentFragment = new ContentFragment()
;
    Bundle args = new Bundle();
    args.putString("text", menuLists.get(position).getIconN
ame());
    contentFragment.setArguments(args);
    FragmentManager fm = getSupportFragmentManager();
    fm.beginTransaction().replace(R.id.ly_content,contentFr
agment).commit();
    drawer_layout.closeDrawer(list_left_drawer);
}
}
```

代码很简单，就不多说了~

8.3.3 Paint API之—— Mask...

8.3.4 Paint API之—— Xferm...

8.3.5 Paint API之—— Xferm...

8.3.6 Paint API之—— Xferm...

8.3.7 Paint API之—— Xferm...

8.3.8 Paint API之—— Xferm...

8.3.9 Paint API之—— Color...

8.3.10 Paint API之—— Colo...

8.3.11 Paint API之—— Colo...

8.3.12 Paint API之—— Path...

8.3.13 Paint API之—— Sha...

8.3.14 Paint几个枚举/常量值...

8.3.15 Paint API之——Type...

8.3.16 Canvas API详解(Part 1)

8.3.17 Canvas API详解(Part...

8.3.18 Canvas API详解(Part...

8.4.1 Android动画合集之帧...

8.4.2 Android动画合集之补...

8.4.3 Android动画合集之属...

8.4.4 Android动画合集之属...

9.1 使用SoundPool播放音...

9.2 MediaPlayer播放音频与...

9.3 使用Camera拍照

9.4 使用MediaRecord录音

10.1 TelephonyManager(电...

10.2 SmsManager(短信管理...

10.3 AudioManager(音频管...

10.4 Vibrator(振动器)

10.5 AlarmManager(闹钟服务)

10.6 PowerManager(电源服...

10.7 WindowManager(窗口...

10.8 LayoutInflater(布局服务)

10.9 WallpaperManager(壁...

10.10 传感器专题(1)——相...

10.11 传感器专题(2)——方...

10.12 传感器专题(3)——加...

10.12 传感器专题(4)——其...

10.14 Android GPS初涉

11.0 《2015最新Android基...

示例2.左右两个侧滑菜单的实现

嗯，不知道你有没有发现，从上面的DrawerLayout的布局，我们大概可以猜到，DrawerLayout 最多由三个部分组成，中间的内容部分，左边的侧滑菜单部分，右边的侧滑菜单部分组成！下面我们来写一个带有两个侧滑菜单的示例！

运行效果图：



代码实现：

首先我们创建两个Fragment以及对应的布局，他们分别是左右侧滑菜单！

左边Fragment：

布局：**fg_left.xml**，这里就用了一个图片而以，点击后弹出一个新的Activity；当然你可以根据自己的需求进行扩展！

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    android:orientation="vertical" android:layout_width="match_
parent"
    android:layout_height="match_parent">
```

```

<ImageView
    android:id="@+id/img_bg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@mipmap/bg_menu_left"/>

</LinearLayout>

```

对应的LeftFragment.java :

```

/**
 * Created by Jay on 2015/10/9 0009.
 */
public class LeftFragment extends Fragment{

    private DrawerLayout drawer_layout;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_left, containe
r, false);
        ImageView img_bg = (ImageView) view.findViewById(R.id.i
mg_bg);
        img_bg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                getActivity().startActivity(new Intent(getActiv
ity(),OtherActivity.class));
                drawer_layout.closeDrawer(Gravity.START);
            }
        });
        return view;
    }

    //暴露给Activity, 用于传入DrawerLayout, 因为点击后想关掉DrawerL
ayout
    public void setDrawerLayout(DrawerLayout drawer_layout){
        this.drawer_layout = drawer_layout;
    }
}

```

右面的Fragment :

布局就三个按钮, 点击后替换中间部分的Fragment, 布局fg_right.xml代码如下 :

```

<?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="match_parent"
android:background="#2F9AF2"
android:gravity="center"
android:orientation="vertical">

<Button
    android:id="@+id/btn_one"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="菜单项一" />

<Button
    android:id="@+id/btn_two"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="菜单项二" />

<Button
    android:id="@+id/btn_three"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="菜单项三" />

</LinearLayout>

```

然后对应的是**RightFragment.java** :

```

/**
 * Created by Jay on 2015/10/9 0009.
 */
public class RightFragment extends Fragment implements View.OnClickListener{

    private DrawerLayout drawer_layout;
    private FragmentManager fManager;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_right, contain
er, false);
        view.findViewById(R.id.btn_one).setOnClickListener(this
);
        view.findViewById(R.id.btn_two).setOnClickListener(this
);
        view.findViewById(R.id.btn_three).setOnClickListener(th
is);
        fManager = getActivity().getSupportFragmentManager();
        return view;
    }

    @Override
    public void onClick(View v) {

```

```

        switch (v.getId()) {
            case R.id.btn_one:
                ContentFragment cFragment1 = new ContentFragmen
t("1.点击了右侧菜单项一",R.color.blue);
                fragmentManager.beginTransaction().replace(R.id.fly_co
ntent,cFragment1).commit();
                drawer_layout.closeDrawer(Gravity.END);
                break;
            case R.id.btn_two:
                ContentFragment cFragment2 = new ContentFragmen
t("2.点击了右侧菜单项二",R.color.red);
                fragmentManager.beginTransaction().replace(R.id.fly_co
ntent,cFragment2).commit();
                drawer_layout.closeDrawer(Gravity.END);
                break;
            case R.id.btn_three:
                ContentFragment cFragment3 = new ContentFragmen
t("3.点击了右侧菜单项三",R.color.yellow);
                fragmentManager.beginTransaction().replace(R.id.fly_co
ntent,cFragment3).commit();
                drawer_layout.closeDrawer(Gravity.END);
                break;
        }
    }

    public void setDrawerLayout(DrawerLayout drawer_layout) {
        this.drawer_layout = drawer_layout;
    }
}

```

另外还有一个中间部分填充的ContentFragment，布局：**fg_content.xml**如下：

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/r
es/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/tv_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:textSize="25sp" />

</RelativeLayout>

```

ContentFragment.java :


```
public class ContentFragment extends Fragment {

    private TextView tv_content;
    private String strContent;
    private int bgColor;

    public ContentFragment(String strContent,int bgColor) {
        this.strContent = strContent;
        this.bgColor = bgColor;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_content, conta
iner, false);
        view.setBackgroundColor(getResources().getColor(bgColor
));
        tv_content = (TextView) view.findViewById(R.id.tv_conte
nt);
        tv_content.setText(strContent);
        return view;
    }
}
```

编写好以后，就到我们的Activity的布局了以及Activity的代码了：在此之前我们还需要些一个顶部条形栏的布局：

view_topbar.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/r
es/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#DCDEDB">

    <Button
        android:id="@+id/btn_right"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_centerVertical="true"
        android:layout_alignParentRight="true"
        android:background="@drawable/btn_selector"/>

</RelativeLayout>
```

然后是activity_main.xml :

```
<android.support.v4.widget.DrawerLayout xmlns:android="http://s
chemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <include

        android:id="@+id/topbar"
        layout="@layout/view_topbar"
        android:layout_width="wrap_content"
        android:layout_height="48dp" />

    <FrameLayout

        android:id="@+id/fly_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>

<fragment

    android:id="@+id/fg_left_menu"
    android:name="jay.com.drawerlayoutdemo2.LeftFragment"
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:tag="LEFT"
    tools:layout="@layout/fg_left" />

<fragment

    android:id="@+id/fg_right_menu"
    android:name="jay.com.drawerlayoutdemo2.RightFragment"
    android:layout_width="100dp"
    android:layout_height="match_parent"
    android:layout_gravity="end"
    android:tag="RIGHT"
    tools:layout="@layout/fg_right" />

</android.support.v4.widget.DrawerLayout>
```

最后是MainActivity.java :

```
public class MainActivity extends AppCompatActivity implements
View.OnClickListener {

    private DrawerLayout drawer_layout;
    private FrameLayout fly_content;
    private View topbar;
    private Button btn_right;
    private RightFragment fg_right_menu;
    private LeftFragment fg_left_menu;
```

```
private FragmentManager fManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    fManager = getSupportFragmentManager();
    fg_right_menu = (RightFragment) fManager.findFragmentById(R.id.fg_right_menu);
    fg_left_menu = (LeftFragment) fManager.findFragmentById(R.id.fg_left_menu);
    initView();
}

private void initView() {
    drawer_layout = (DrawerLayout) findViewById(R.id.drawer_layout);
    fly_content = (FrameLayout) findViewById(R.id.fly_content);
    topbar = findViewById(R.id.topbar);
    btn_right = (Button) topbar.findViewById(R.id.btn_right);
    btn_right.setOnClickListener(this);

    //设置右面的侧滑菜单只能通过编程来打开
    drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED, Gravity.END);

    drawer_layout.setDrawerListener(new DrawerLayout.DrawerListener() {
        @Override
        public void onDrawerSlide(View view, float v) {

        }

        @Override
        public void onDrawerOpened(View view) {

        }

        @Override
        public void onDrawerClosed(View view) {
            drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED, Gravity.END);
        }

        @Override
        public void onDrawerStateChanged(int i) {

        }
    });
}
```

```
fg_right_menu.setDrawerLayout(drawer_layout);
fg_left_menu.setDrawerLayout(drawer_layout);
}

@Override
public void onClick(View v) {
    drawer_layout.openDrawer(Gravity.RIGHT);
    drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED,
                                     Gravity.END);    //解除锁定
}
}
```

好的，至此就大功告成了~，呼呼，下面说下看代码时可能会有的疑惑：

1. drawer_layout.openDrawer(Gravity.END);

这句是设置打开的哪个菜单START代表左边，END代表右边

2.

drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED,Gravity.END);

锁定右面的侧滑菜单，不能通过手势关闭或者打开，只能通过代码打开！即调用openDrawer方法！接着

drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED,Gravity.END);

解除锁定状态，即可以通过手势关闭侧滑菜单 最后在drawer关闭的时候调用：

drawer_layout.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED, Gravity.END); 再次锁定右边的侧滑菜单！

3. 布局代码中的Tag属性的作用？ 答：这里没用到，在重写DrawerListener的onDrawerSlide方法时，我们可以通过他的第一个参数drawerView，调用drawerView.getTag().equals("START")判断触发菜单事件的是哪个 菜单！然后可以进行对应的操作！

3.代码示例下载

[DrawerLayoutDemo.zip](#)

[DrawerLayoutDemo2.zip](#)

本节小结：

好的，本节给大家介绍了官方的侧滑控件DrawerLayout的基本用法，使用起来非常的方便！当然这里仅仅是简单的使用演示，另外看到弘扬大神写过一篇：[Android DrawerLayout 高仿QQ5.2双向侧滑菜单](#) 有兴趣可以看看，如果看完本节的内容，相信你看起来不会怎么吃力~好的！

本节就到这里，跟UI控件这一章说拜拜了~下一章我们开始绘图与动

画了，为我们进阶部分的自定义控件系列打基础！





在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [HTML ISO-8859-1](#)
- [HTML 实体符号](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)

最新更新

- [Shell 编程快速入门](#)
- [Shell 文件包含](#)
- [Shell 输入/输出...](#)
- [Shell printf 命令](#)
- [Shell 基本运算符](#)
- [PHP接收并压缩图...](#)
- [Shell 数组](#)

站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

关注微信



Copyright © 2013-2015 菜鸟教程 **runoob.com** All Rights Reserved. 备案号：闽ICP备15012807号-1