

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

## 5.2.5 Fragment实例精讲—— 新闻(购物)类App列表 Fragment的简单实现

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：

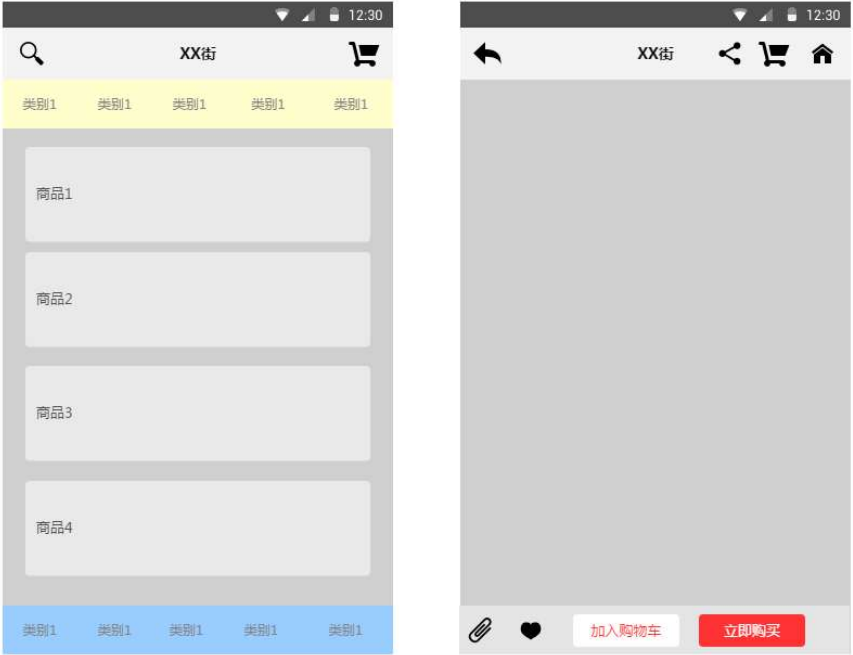
相信大家对点击列表，然后进入详情这种App并不陌生吧，在购物类App和新闻类App中最为常见：下面我们简单来讲一下流程逻辑！

### 1.逻辑流程讲解：

刚好公司测试妹子的测试机上装了楚楚街9块9的APP，呵呵，直接就照这个来研究吧：



嘿嘿，市面上很多APP都是这种样子的，而这个可以用我们学到的Fragment来实现：可能gif动画看不清，笔者用界面原型工具画个大概吧：

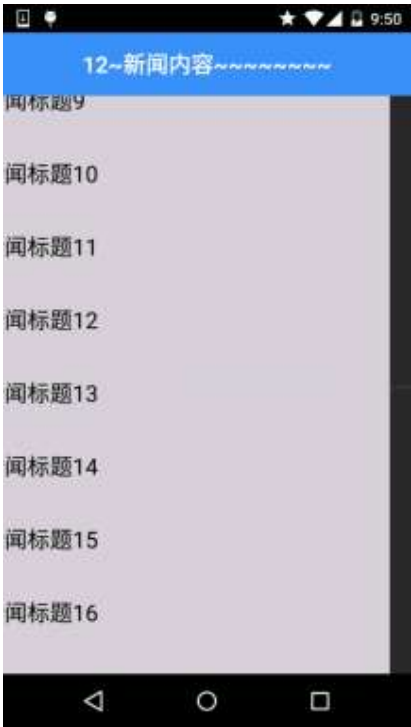


大概就这样，中间区域是一个布局容器，一般是FrameLayout，然后将一个Fragment replace 到这个容器中或者add也行，而这个Fragment中有一个listview，当我们点击这个ListView中的一项，中间容器中的Fragment就会被replace成对应详细信息的Fragment所替代，如果我们只是replace的话，就不会保存第一个Fragment的状态，用户又得从头开始浏览，这肯定是很不方便的，这里我们可以 通过Fragment栈的addtoBackStack和popBackStack来解决这个问题！当replace的同时，我们将被替换 的Fragment添加到stack中，当用户点击回退按钮时，调用popBackStack弹出栈，具体实现见下述代码 示例！

## 2.代码示例：简单新闻类APP列表和内容切换的实现

运行效果图：

- 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
- 2.4.7ListView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 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewFlipper(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



实现代码：

Step 1:先把两个Fragment以及Activity的布局实现了

fg\_newlist.xml:

```
<?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="@color/white"
    android:orientation="horizontal">

    <ListView
        android:id="@+id/list_news"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

fg\_context.xml:

```
<?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:orientation="vertical">

    <TextView
        android:id="@+id/txt_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
```

- 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

```
        android:textColor="@color/blue"
        android:textSize="20sp" />
```

```
</LinearLayout>
```

### activity\_main.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt_title"
        android:layout_width="match_parent"
        android:layout_height="56dp"
        android:background="@color/blue"
        android:textColor="@color/white"
        android:text="新闻列表"
        android:textSize="20sp"
        android:textStyle="bold"
        android:gravity="center"/>

    <FrameLayout
        android:id="@+id/fl_content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/txt_title"/>

</RelativeLayout>
```

### Step 2:实现我们的业务Bean类和自定义BaseAdapter类：

#### Data.java:

```
/**
 * Created by Jay on 2015/9/6 0006.
 */
public class Data {

    private String new_title;
    private String new_content;

    public Data() {}

    public Data(String new_title, String new_content) {
        this.new_title = new_title;
        this.new_content = new_content;
    }
}
```

- 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...

```
public String getNew_title() {
    return new_title;
}

public String getNew_content() {
    return new_content;
}

public void setNew_title(String new_title) {
    this.new_title = new_title;
}

public void setNew_content(String new_content) {
    this.new_content = new_content;
}
}
```

### MyAdapter.java:

```
/**
 * Created by Jay on 2015/9/6 0006.
 */
public class MyAdapter extends BaseAdapter{

    private List<Data> mData;
    private Context mContext;

    public MyAdapter(List<Data> mData, Context mContext) {
        this.mData = mData;
        this.mContext = mContext;
    }

    @Override
    public int getCount() {
        return mData.size();
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder viewHolder;
        if(convertView == null){
            convertView = LayoutInflater.from(mContext).inflate
```

```

(R.layout.list_item,parent,false);
    viewHolder = new ViewHolder();
    viewHolder.txt_item_title = (TextView) convertView.
findViewById(R.id.txt_item_title);
    convertView.setTag(viewHolder);
    }else{
        viewHolder = (ViewHolder) convertView.getTag();
    }
    viewHolder.txt_item_title.setText(mData.get(position).getNew_title());
    return convertView;
}

private class ViewHolder{
    TextView txt_item_title;
}
}

```

### Step 3:MainActivity的实现

#### MainActivity.java:

```

public class MainActivity extends AppCompatActivity {

    private TextView txt_title;
    private FrameLayout fl_content;
    private Context mContext;
    private ArrayList<Data> datas = null;
    private FragmentManager fManager = null;
    private long exitTime = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mContext = MainActivity.this;
        fManager = getFragmentManager();
        bindViews();

        datas = new ArrayList<Data>();
        for (int i = 1; i <= 20; i++) {
            Data data = new Data("新闻标题" + i, i + "~新闻内容~~~");
            datas.add(data);
        }
        NewListFragment nlFragment = new NewListFragment(fManager, datas);
        FragmentTransaction ft = fManager.beginTransaction();
        ft.replace(R.id.fl_content, nlFragment);
        ft.commit();
    }
}

```

```

private void bindViews() {
    txt_title = (TextView) findViewById(R.id.txt_title);
    fl_content = (FrameLayout) findViewById(R.id.fl_content);
};

//点击回退键的处理：判断Fragment栈中是否有Fragment
//没，双击退出程序，否则像是Toast提示
//有，popBackStack弹出栈
@Override
public void onBackPressed() {
    if (fManager.getBackStackEntryCount() == 0) {
        if ((System.currentTimeMillis() - exitTime) > 2000)
        {
            Toast.makeText(getApplicationContext(), "再按一次退出程序",
                Toast.LENGTH_SHORT).show();
            exitTime = System.currentTimeMillis();
        } else {
            super.onBackPressed();
        }
    } else {
        fManager.popBackStack();
        txt_title.setText("新闻列表");
    }
}
}

```

#### Step 4:列表Fragment的实现：

##### NewListFragment.java：

```

package com.jay.fragmentdemo4;

import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.TextView;

import java.util.ArrayList;

/**
 * Created by Jay on 2015/9/6 0006.
 */

```

```
public class NewListFragment extends Fragment implements AdapterView.OnItemClickListener {
    private FragmentManager fManager;
    private ArrayList<Data> datas;
    private ListView list_news;

    public NewListFragment(FragmentManager fManager, ArrayList<Data> datas) {
        this.fManager = fManager;
        this.datas = datas;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_newlist, container, false);
        list_news = (ListView) view.findViewById(R.id.list_news);

        MyAdapter myAdapter = new MyAdapter(datas, getActivity());
        list_news.setAdapter(myAdapter);
        list_news.setOnItemClickListener(this);
        return view;
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        FragmentTransaction fTransaction = fManager.beginTransaction();
        NewContentFragment ncFragment = new NewContentFragment();

        Bundle bd = new Bundle();
        bd.putString("content", datas.get(position).getNew_content());
        ncFragment.setArguments(bd);
        //获取Activity的控件
        TextView txt_title = (TextView) getActivity().findViewById(R.id.txt_title);
        txt_title.setText(datas.get(position).getNew_content());
        ;

        //加上Fragment替换动画
        fTransaction.setCustomAnimations(R.anim.fragment_slide_left_enter, R.anim.fragment_slide_left_exit);
        fTransaction.replace(R.id.fl_content, ncFragment);
        //调用addToBackStack将Fragment添加到栈中
        fTransaction.addToBackStack(null);
        fTransaction.commit();
    }
}
```



## Step 5:内容Fragment的实现：

### NewContentFragment.java:

```
/**
 * Created by Jay on 2015/9/6 0006.
 */
public class NewContentFragment extends Fragment {

    NewContentFragment() {

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fg_content, conta
iner, false);
        TextView txt_content = (TextView) view.findViewById(R.i
d.txt_content);
        //getArgument获取传递过来的Bundle对象
        txt_content.setText(getArguments().getString("content")
);
        return view;
    }

}
```

代码很简单，就不慢慢解释了~

## 3.代码下载

FragmentDemo5.zip : [下载 FragmentDemo5.zip](#)

## 本节小结：

因为时间的关系，并没有详细的去做过多的讲解，示例代码也很简单，方便各位初学者理解！如果要用到实际项目中还需要对此进行一番修改~！好的，本节就到这里，谢谢~

← 5.2.4 Fragment实例精讲——底部导航栏+ViewPager滑动切换页面

6.1 数据存储与访问之——文件存储读写 →

大家都在玩手机而我却用手机学英语

邮箱订阅《每日英语》  
每天坚持五分钟，随时随地学英语！

立即行动



	<ul style="list-style-type: none"><li>· HTML 实例</li><li>· CSS 实例</li><li>· JavaScript 实例</li><li>· Ajax 实例</li><li>· jQuery 实例</li><li>· XML 实例</li><li>· Java 实例</li></ul>	<ul style="list-style-type: none"><li>· HTML 字符集设置</li><li>· HTML ASCII 字符集</li><li>· HTML ISO-8859-1</li><li>· HTML 实体符号</li><li>· HTML 拾色器</li><li>· JSON 格式化工具</li></ul>	<ul style="list-style-type: none"><li>· Vue.js 入门教程</li><li>· CSS 下拉菜单</li><li>· WordPress Nginx...</li><li>· TIOBE 2015 年12...</li><li>· Swift 正式开源</li><li>· PHP 7 正式发布</li><li>· Shell 编程快速入门</li></ul>	<ul style="list-style-type: none"><li>· 意见反馈</li><li>· 免责声明</li><li>· 关于我们</li><li>· 文章归档</li></ul>
				<div>关注微信</div> <div></div> <div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div>