

# Hongyang

生命不息，奋斗不止，万事起于忽微，量变引起质变

目录视图

## 个人资料



鸿洋\_

关注

发私信



访问：9628568次

积分：40374

等级：BLOG > 8

排名：第64名

原创：188篇

转载：0篇

译文：6篇

评论：11452条

## 我的微信公众号

点击直达推送文章汇总

长期为您推荐优秀博文、开源项目、视频等，进入还有好玩的等着您，欢迎扫一扫。



## 联系方式

新动态

给我写信

QQ群：

429757068

264950424

463081660

请勿重复加群，Thx

## 文章分类

【Android 5.x】 (11)

【公告】博客专栏旧貌换新颜    【hot】直播技术精选    主流编程语言图谱之二

## Android Fragment 真正的完全解析（下）

标签：Android   Fragment   DialogFragment   MenuItem

2014-07-21 09:09

183412人阅读

评论(1)

分类：    【Android 源码解析】（28）    【android 进阶之路】（64）

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

转载请标明出处：<http://blog.csdn.net/Imj623565791/article/details/37992017>

上篇博客中已经介绍了Fragment产生原因，以及一些基本的用法和各种API，如果你还不了解，请看：[Android Fragment（上）](#)。

本篇将介绍上篇博客提到的：如何管理Fragment回退栈，Fragment如何与Activity交互，Fragment与Activity交互的多种方式，Fragment的用处，使用Fragment创建对话框，如何与ActionBar，MenuItem集成等~~

### 1、管理Fragment回退栈

类似与Android系统为Activity维护一个任务栈，我们也可以通过Activity维护一个回退栈来保存每一次任务发生。当用户点击后退按钮时，将看到上一轮的保存的Fragment。一旦Fragment被移出回退栈，则退出当前Activity。

看这样一个效果图：

- 【Android 精彩案例】 (37)
- 【Android 源码解析】 (29)
- 【Android 自定义控件实战】 (29)
- 【Android 自定义控件之起步】 (7)
- 【Android 快速开发】 (12)
- 【Android 原生开发游戏】 (3)
- 【Java 并发专题】 (15)
- 【android 进阶之路】 (65)
- 【Java 设计模式】 (10)
- 【Android 百度地图】 (4)
- 【html5 css3精彩案例】 (14)
- 【Android github 控件】 (10)
- 【Android 基础】 (16)
- 【Javascript】 (9)
- 【rabbitMQ 用法】 (5)
- 【Android微知识点】 (4)

友情链接

- 郭霖的博客
- 夏安明的博客
- 任玉刚的博客
- 元斌的博客
- 敬佩的孔老师
- foruok的订阅号程序视界
- OpenCV大神shiter
- 专为Android程序员的导航
- 泡在网上的日子

博客专栏

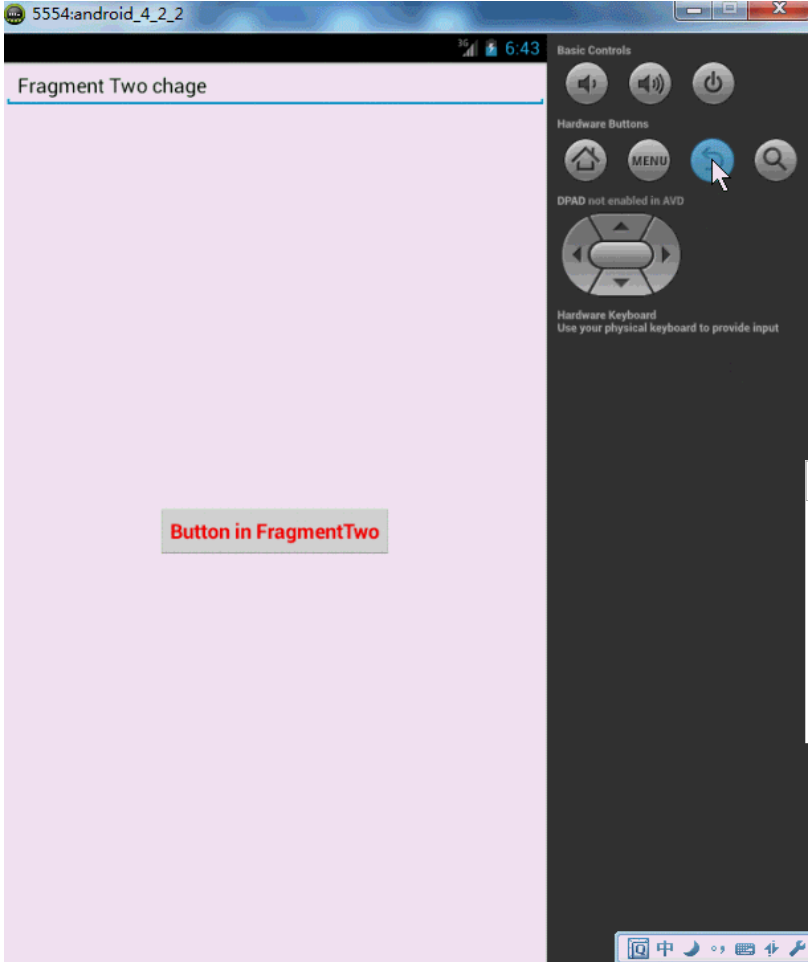
-  HTML5 & CSS3 实战  
文章：11篇  
阅读：134533
-  设计模式融入生活  
文章：10篇  
阅读：82126
-  Android 精彩案例  
文章：67篇  
阅读：4065558

阅读排行

- Android Https相关完全解析 ... (1523455)
- Android Fragment 真正的完... (470383)
- Android RecyclerView 使用... (349532)
- Android OkHttp完全解析 是... (265635)
- Android 自定义View (一) (195645)
- Android 属性动画 ( Property... (189638)
- Android Fragment 真正的完... (183244)
- Android 屏幕适配方案 (140236)
- Android 手把手教您自定义Vi... (139640)
- Android 自定义RecyclerVie... (124154)

最新评论

- Android SwipeRefreshLayout 官方下拉...



点击第一个按钮，切换到第二个界面，点击第二个按钮，切换到第三个界面，然后点击Back键依次回退。这像不像初学转，当然了，这里肯定不是，不然我就跪了。这里是Fragment实现的，用户点击Back，实际是Fragment回退栈不断的如何添加一个Fragment事务到回退栈：

FragmentTransaction.addToBackStack(String)

下面讲解代码：很明显一共是3个Fragment和一个Activity.

先看Activity的布局文件：

```
[html]
01. <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
02.     xmlns:tools="http://schemas.android.com/tools"
03.     android:layout_width="match_parent"
04.     android:layout_height="match_parent" >
05.
06.     <FrameLayout
07.         android:id="@+id/id_content"
08.         android:layout_width="fill_parent"
09.         android:layout_height="fill_parent" >
10.     </FrameLayout>
11.
12. </RelativeLayout>
```

不同的Fragment就在这个FrameLayout中显示。

MainActivity.java

```
[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Activity;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.view.Window;
08.
09. public class MainActivity extends Activity
10. {
11.
12.
```

- lilei76456967 : hyman,没有看到上面加载条的效果呢
- Android NestedScrolling机制完全解析 ...  
IT\_peng : hongyang 大神你好, RecyclerView View List 数据少的情况下 下面会有一篇...
- Android RecyclerView 使用完全解析 体...  
游鑫 : https://github.com/youxin11544/XRecyclerView RecyclerView
- Android HandlerThread 完全解析  
qq\_36021289 : @z995204127:handleMessage方法如果不是运行在主线程,那怎么更新你的textv...
- Android OkHttp完全解析 是时候来了解...  
Stone\_birds : get请求一直是一下这样的错误:E/TAG: =====request'log'=====E...
- Android NestedScrolling机制完全解析 ...  
qq\_34379889 : 大神,不能把把内容区换成ListView或者gridView这种更常见的View吗,我使用ListV...
- Android NestedScrolling机制完全解析 ...  
Notzuonotdied : 谢谢大神~
- 神奇的Android Studio Template  
R3lish : https://github.com/MontazzeStudio/android-studio-t...
- Android RecyclerView 使用完全解析 体...  
qq\_35768590 : 写得真是太好了,通俗易懂,谢谢!
- HTML5 CSS3专题 纯CSS打造相册效果  
十一期王海鹏 : 图片浏览实例

2015博客之星评选

统计



```
13.     @Override
14.     protected void onCreate(Bundle savedInstanceState)
15.     {
16.         super.onCreate(savedInstanceState);
17.         requestWindowFeature(Window.FEATURE_NO_TITLE);
18.         setContentView(R.layout.activity_main);
19.
20.         FragmentManager fm = getFragmentManager();
21.         FragmentTransaction tx = fm.beginTransaction();
22.         tx.add(R.id.id_content, new FragmentOne(), "ONE");
23.         tx.commit();
24.     }
25.
26. }
```

很简单，直接将FragmentOne添加到布局文件中的FrameLayout中，注意这里并没有调用FragmentTransaction.add  
因为我不喜欢在当前显示时，点击Back键出现白板。而是正确的相应Back键，即退出我们的Activity。

下面是FragmentOne

```
[java] C P
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.view.LayoutInflater;
08. import android.view.View;
09. import android.view.View.OnClickListener;
10. import android.view.ViewGroup;
11. import android.widget.Button;
12.
13. public class FragmentOne extends Fragment implements OnClickListener
14. {
15.
16.     private Button mBtn;
17.
18.     @Override
19.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
20.         Bundle savedInstanceState)
21.     {
22.         View view = inflater.inflate(R.layout.fragment_one, container, false);
23.         mBtn = (Button) view.findViewById(R.id.id_fragment_one_btn);
24.         mBtn.setOnClickListener(this);
25.         return view;
26.     }
27.
28.     @Override
29.     public void onClick(View v)
30.     {
31.         FragmentTwo fTwo = new FragmentTwo();
32.         FragmentManager fm = getFragmentManager();
33.         FragmentTransaction tx = fm.beginTransaction();
34.         tx.replace(R.id.id_content, fTwo, "TWO");
35.         tx.addToBackStack(null);
36.         tx.commit();
37.     }
38.
39.
40. }
```



我们在点击FragmentOne中的按钮时，使用了replace方法，如果你看了前一篇博客，一定记得replace是remove和ac  
加事务到回退栈，前一个Fragment实例会被销毁。这里很明显，我们调用tx.addToBackStack(null);将当前的事务添加  
FragmentOne实例不会被销毁，但是视图层次依然会被销毁，即会调用onDestoryView和onCreateView，证据就是：  
我们在跳转前在文本框输入的内容，在用户Back得到第一个界面的时候不见了。

接下来FragmentTwo

```
[java] C P
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.view.LayoutInflater;
```

```

08. import android.view.View;
09. import android.view.View.OnClickListener;
10. import android.view.ViewGroup;
11. import android.widget.Button;
12.
13. public class FragmentTwo extends Fragment implements OnClickListener
14. {
15.
16.     private Button mBtn ;
17.     @Override
18.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
19.         Bundle savedInstanceState)
20.     {
21.         View view = inflater.inflate(R.layout.fragment_two, container, false);
22.         mBtn = (Button) view.findViewById(R.id.id_fragment_two_btn);
23.         mBtn.setOnClickListener(this);
24.         return view ;
25.     }
26.     @Override
27.     public void onClick(View v)
28.     {
29.         FragmentThree fThree = new FragmentThree();
30.         FragmentManager fm = getFragmentManager();
31.         FragmentTransaction tx = fm.beginTransaction();
32.         tx.hide(this);
33.         tx.add(R.id.id_content , fThree, "THREE");
34.         // tx.replace(R.id.id_content, fThree, "THREE");
35.         tx.addToBackStack(null);
36.         tx.commit();
37.     }
38.
39.
40. }

```

微信关注我的公众号



这里点击时，我们没有使用replace，而是先隐藏了当前的Fragment，然后添加了FragmentThree的实例，最后将事务的目的是为了给大家提供一种方案：如果不希望视图重绘该怎么做，请再次仔细看效果图，我们在FragmentTwo的EditBack回来时，数据还在~~~

最后FragmentThree就是简单的Toast了：

```

[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.os.Bundle;
05. import android.view.LayoutInflater;
06. import android.view.View;
07. import android.view.View.OnClickListener;
08. import android.view.ViewGroup;
09. import android.widget.Button;
10. import android.widget.Toast;
11.
12. public class FragmentThree extends Fragment implements OnClickListener
13. {
14.
15.     private Button mBtn;
16.
17.     @Override
18.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
19.         Bundle savedInstanceState)
20.     {
21.         View view = inflater.inflate(R.layout.fragment_three, container, false);
22.         mBtn = (Button) view.findViewById(R.id.id_fragment_three_btn);
23.         mBtn.setOnClickListener(this);
24.         return view;
25.     }
26.
27.     @Override
28.     public void onClick(View v)
29.     {
30.         Toast.makeText(getActivity(), " i am a btn in Fragment three",
31.             Toast.LENGTH_SHORT).show();
32.     }
33.
34. }

```

好了，经过上面的介绍，应该已经知道Fragment回退栈是怎么一回事了，以及hide，replace等各自的应用的场景。

这里极其注意一点：上面的整体代码不具有任何参考价值，纯粹为了显示回退栈，在后面讲解了Fragment与Activity通信！

## 2、Fragment与Activity通信

因为所有的Fragment都是依附于Activity的，所以通信起来并不复杂，大概归纳为：

- a、如果你Activity中包含自己管理的Fragment的引用，可以通过引用直接访问所有的Fragment的public方法
- b、如果Activity中未保存任何Fragment的引用，那么没关系，每个Fragment都有一个唯一的TAG或者ID,可以通过getFragmentManager.findFragmentByTag()或者findFragmentById()获得任何Fragment实例，然后进行操作。
- c、在Fragment中可以通过getActivity得到当前绑定的Activity的实例，然后进行操作。

注：如果在Fragment中需要Context，可以通过调用getActivity(),如果该Context需要在Activity被销毁后还存在，则作getActivity().getApplicationContext()。

## 3、Fragment与Activity通信的最佳实践

因为要考虑Fragment的重复使用，所以必须降低Fragment与Activity的耦合，而且Fragment操作应该由它的管理者Activity来决定。

下面我通过两种方式的代码，分别重构，FragmentOne和FragmentTwo的点击事件，以及首先看FragmentOne

```
[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.os.Bundle;
05. import android.view.LayoutInflater;
06. import android.view.View;
07. import android.view.View.OnClickListener;
08. import android.view.ViewGroup;
09. import android.widget.Button;
10.
11. public class FragmentOne extends Fragment implements OnClickListener
12. {
13.     private Button mBtn;
14.
15.     /**
16.      * 设置按钮点击的回调
17.      * @author zhy
18.      */
19.     public interface FOneBtnClickListener
20.     {
21.         void onFOneBtnClick();
22.     }
23.
24.
25.     @Override
26.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
27.         Bundle savedInstanceState)
28.     {
29.         View view = inflater.inflate(R.layout.fragment_one, container, false);
30.         mBtn = (Button) view.findViewById(R.id.id_fragment_one_btn);
31.         mBtn.setOnClickListener(this);
32.         return view;
33.     }
34.
35.     /**
36.      * 交给宿主Activity处理，如果它希望处理
37.      */
38.     @Override
39.     public void onClick(View v)
40.     {
41.         if (getActivity() instanceof FOneBtnClickListener)
42.         {
43.             ((FOneBtnClickListener) getActivity()).onFOneBtnClick();
44.         }
45.     }
46.
47. }
```

可以看到现在的FragmentOne不和任何Activity耦合，任何Activity都可以使用；并且我们声明了一个接口，来回调其点击事件的Activity实现此接口就即可。可以看到我们在onClick中首先判断了当前绑定的Activity是否实现了该接口，如果

微信关注我的公众号



再看FragmentTwo

```

[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.os.Bundle;
05. import android.view.LayoutInflater;
06. import android.view.View;
07. import android.view.View.OnClickListener;
08. import android.view.ViewGroup;
09. import android.widget.Button;
10.
11. public class FragmentTwo extends Fragment implements OnClickListener
12. {
13.
14.
15.     private Button mBtn ;
16.
17.     private FTwoBtnClickListener fTwoBtnClickListener ;
18.
19.     public interface FTwoBtnClickListener
20.     {
21.         void onFTwoBtnClick();
22.     }
23.     //设置回调接口
24.     public void setFTwoBtnClickListener(FTwoBtnClickListener fTwoBtnClickListener)
25.     {
26.         this.fTwoBtnClickListener = fTwoBtnClickListener;
27.     }
28.     @Override
29.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
30.         Bundle savedInstanceState)
31.     {
32.         View view = inflater.inflate(R.layout.fragment_two, container, false);
33.         mBtn = (Button) view.findViewById(R.id.id_fragment_two_btn);
34.         mBtn.setOnClickListener(this);
35.         return view ;
36.     }
37.     @Override
38.     public void onClick(View v)
39.     {
40.         if(fTwoBtnClickListener != null)
41.         {
42.             fTwoBtnClickListener.onFTwoBtnClick();
43.         }
44.     }
45.
46. }

```

微信关注我的公众号



与FragmentOne极其类似，但是我们提供了setListener这样的方法，意味着Activity不仅需要实现该接口，还必须显示mFTwo.setFTwoBtnClickListener(this)。

最后看Activity：

```

[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Activity;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.view.Window;
08.
09. import com.zhy.zhy_fragments.FragmentOne.FOneBtnClickListener;
10. import com.zhy.zhy_fragments.FragmentTwo.FTwoBtnClickListener;
11.
12. public class MainActivity extends Activity implements FOneBtnClickListener,
13.     FTwoBtnClickListener
14. {
15.
16.     private FragmentOne mFOne;
17.     private FragmentTwo mFTwo;
18.     private FragmentThree mFThree;
19.
20.     @Override
21.     protected void onCreate(Bundle savedInstanceState)
22.     {
23.         super.onCreate(savedInstanceState);
24.         requestWindowFeature(Window.FEATURE_NO_TITLE);

```

```

25.         setContentView(R.layout.activity_main);
26.
27.         mFOne = new FragmentOne();
28.         FragmentManager fm = getFragmentManager();
29.         FragmentTransaction tx = fm.beginTransaction();
30.         tx.add(R.id.id_content, mFOne, "ONE");
31.         tx.commit();
32.     }
33.
34.     /**
35.      * FragmentOne 按钮点击时的回调
36.      */
37.     @Override
38.     public void onFOneBtnClick()
39.     {
40.
41.         if (mFTwo == null)
42.         {
43.             mFTwo = new FragmentTwo();
44.             mFTwo.setFTwoBtnClickListener(this);
45.         }
46.         FragmentManager fm = getFragmentManager();
47.         FragmentTransaction tx = fm.beginTransaction();
48.         tx.replace(R.id.id_content, mFTwo, "TWO");
49.         tx.addToBackStack(null);
50.         tx.commit();
51.     }
52.
53.     /**
54.      * FragmentTwo 按钮点击时的回调
55.      */
56.     @Override
57.     public void onFTwoBtnClick()
58.     {
59.         if (mFThree == null)
60.         {
61.             mFThree = new FragmentThree();
62.
63.         }
64.         FragmentManager fm = getFragmentManager();
65.         FragmentTransaction tx = fm.beginTransaction();
66.         tx.hide(mFTwo);
67.         tx.add(R.id.id_content, mFThree, "THREE");
68.         // tx.replace(R.id.id_content, fThree, "THREE");
69.         tx.addToBackStack(null);
70.         tx.commit();
71.     }
72.
73. }

```

微信关注我的公众号



代码重构结束，与开始的效果一模一样。上面两种通信方式都是值得推荐的，随便选择一种自己喜欢的。这里再提一下Activity可以通过getActivity与findFragmentByTag或者findFragmentById，进行任何操作，甚至在Fragment里面操作是没有特殊理由是绝对不提倡的。Activity担任的是Fragment间类似总线一样的角色，应当由它决定Fragment如何操作能响应Intent打开，但是Activity可以，Activity可以接收Intent，然后根据参数判断显示哪个Fragment。

#### 4、如何处理运行时配置发生变化

运行时配置发生变化，最常见的就是屏幕发生旋转，如果你不知道如何处理屏幕变化可以参考：[Android 屏幕旋转 处理 ProgressDialog 的最佳方案](#)

这里提一下：很多人觉得强制设置屏幕的方向就可以了，但是有一点，当你的应用被至于后台（例如用户点击了home时候，你的应用也会被重新启动。比如上例：如果你把上面的例子你至于FragmentThree界面，然后处于后台状态，长时过home打开时，上面FragmentThree与FragmentOne叠加在一起，这就是因为你的Activity重新启动，在原来的FragmentOne。

好了，下面看一段代码：

Activity:

```

[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Activity;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.view.Window;

```

```
08.
09. public class MainActivity extends Activity
10.
11. {
12.     private FragmentOne mFOne;
13.
14.     @Override
15.     protected void onCreate(Bundle savedInstanceState)
16.     {
17.         super.onCreate(savedInstanceState);
18.         requestWindowFeature(Window.FEATURE_NO_TITLE);
19.         setContentView(R.layout.activity_main);
20.
21.         mFOne = new FragmentOne();
22.         FragmentManager fm = getFragmentManager();
23.         FragmentTransaction tx = fm.beginTransaction();
24.         tx.add(R.id.id_content, mFOne, "ONE");
25.         tx.commit();
26.
27.     }
28.
29. }
```

微信关注我的公众号



## Fragment

```
[java] C P
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.os.Bundle;
05. import android.util.Log;
06. import android.view.LayoutInflater;
07. import android.view.View;
08. import android.view.ViewGroup;
09.
10. public class FragmentOne extends Fragment
11. {
12.     private static final String TAG = "FragmentOne";
13.
14.     @Override
15.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
16.         Bundle savedInstanceState)
17.     {
18.         Log.e(TAG, "onCreateView");
19.         View view = inflater.inflate(R.layout.fragment_one, container, false);
20.         return view;
21.     }
22.
23.     @Override
24.     public void onCreate(Bundle savedInstanceState)
25.     {
26.         // TODO Auto-generated method stub
27.         super.onCreate(savedInstanceState);
28.
29.         Log.e(TAG, "onCreate");
30.     }
31.
32.     @Override
33.     public void onDestroyView()
34.     {
35.         // TODO Auto-generated method stub
36.         super.onDestroyView();
37.         Log.e(TAG, "onDestroyView");
38.     }
39.
40.     @Override
41.     public void onDestroy()
42.     {
43.         // TODO Auto-generated method stub
44.         super.onDestroy();
45.         Log.e(TAG, "onDestroy");
46.     }
47.
48. }
```

很简单的代码，当你运行之后，不断的旋转屏幕，你会发现每旋转一次屏幕，屏幕上就多了一个FragmentOne的实例，多套生命周期的回调。

类似：



```
[html] C 8
01. 07-20 08:18:46.651: E/FragmentOne(1633): onCreate
02. 07-20 08:18:46.651: E/FragmentOne(1633): onCreate
03. 07-20 08:18:46.651: E/FragmentOne(1633): onCreate
04. 07-20 08:18:46.681: E/FragmentOne(1633): onCreateView
05. 07-20 08:18:46.831: E/FragmentOne(1633): onCreateView
06. 07-20 08:18:46.891: E/FragmentOne(1633): onCreateView
```

这是为什么呢，因为当屏幕发生旋转，Activity发生重新启动，默认的Activity中的Fragment也会跟着Activity重新创建时候，本身存在的Fragment会重新启动，然后当执行Activity的onCreate时，又会再次实例化一个新的Fragment，这就那么如何解决呢：

其实通过检查onCreate的参数Bundle savedInstanceState就可以判断，当前是否发生Activity的重新创建：

默认的savedInstanceState会存储一些数据，包括Fragment的实例：通过打印可以看出：

```
[java] C 8
01. 07-
20 08:23:12.952: E/FragmentOne(1782): Bundle[{android:fragments=android.app.Fragment
```

所以，我们简单改一下代码，只有在savedInstanceState==null时，才进行创建Fragment

```
[java] C 8
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Activity;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.util.Log;
08. import android.view.Window;
09.
10. public class MainActivity extends Activity
11. {
12.     private static final String TAG = "FragmentOne";
13.     private FragmentOne mFOne;
14.
15.     @Override
16.     protected void onCreate(Bundle savedInstanceState)
17.     {
18.         super.onCreate(savedInstanceState);
19.         requestWindowFeature(Window.FEATURE_NO_TITLE);
20.         setContentView(R.layout.activity_main);
21.
22.         Log.e(TAG, savedInstanceState+"");
23.
24.         if(savedInstanceState == null)
25.         {
26.             mFOne = new FragmentOne();
27.             FragmentManager fm = getFragmentManager();
28.             FragmentTransaction tx = fm.beginTransaction();
29.             tx.add(R.id.id_content, mFOne, "ONE");
30.             tx.commit();
31.         }
32.     }
33.
34.
35.
36. }
37.
38. }
```

现在无论进行多次旋转都只会会有一个Fragment实例在Activity中。

现在还存在一个问题，就是重新绘制时，Fragment发生重建，原本的数据如何保持？

其实和Activity类似，Fragment也有onSaveInstanceState的方法，在此方法中进行保存数据，然后在onCreate或者onActivityCreated进行恢复都可以。

由于篇幅原因，就不贴测试代码了。

## 5、Fragment与ActionBar和MenuItem集成

Fragment可以添加自己的MenuItem到Activity的ActionBar或者可选菜单中。

微信关注我的公众号



a、在Fragment的onCreate中调用 setHasOptionsMenu(true);

b、然后在Fragment子类中实现onCreateOptionsMenu

c、如果希望在Fragment中处理MenuItem的点击，也可以实现onOptionsItemSelected；当然了Activity也可以直点击事件。

代码：

Fragment

```
[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Fragment;
04. import android.os.Bundle;
05. import android.view.LayoutInflater;
06. import android.view.Menu;
07. import android.view.MenuInflater;
08. import android.view.MenuItem;
09. import android.view.View;
10. import android.view.ViewGroup;
11. import android.widget.Toast;
12.
13. public class FragmentOne extends Fragment
14. {
15.
16.     @Override
17.     public void onCreate(Bundle savedInstanceState)
18.     {
19.         super.onCreate(savedInstanceState);
20.         setHasOptionsMenu(true);
21.     }
22.
23.     @Override
24.     public View onCreateView(LayoutInflater inflater, ViewGroup container,
25.         Bundle savedInstanceState)
26.     {
27.         View view = inflater.inflate(R.layout.fragment_one, container, false);
28.         return view;
29.     }
30.
31.     @Override
32.     public void onCreateOptionsMenu(Menu menu, MenuInflater inflater)
33.     {
34.         inflater.inflate(R.menu.fragment_menu, menu);
35.     }
36.
37.     @Override
38.     public boolean onOptionsItemSelected(MenuItem item)
39.     {
40.         switch (item.getItemId())
41.         {
42.             case R.id.id_menu_fra_test:
43.                 Toast.makeText(getActivity(), "FragmentMenuItem", Toast.LENGTH_SHORT).show();
44.                 break;
45.         }
46.         return true;
47.     }
48.
49. }
```

微信关注我的公众号



Activity

```
[java]
01. package com.zhy.zhy_fragments;
02.
03. import android.app.Activity;
04. import android.app.FragmentManager;
05. import android.app.FragmentTransaction;
06. import android.os.Bundle;
07. import android.util.Log;
08. import android.view.Menu;
09. import android.view.MenuItem;
10. import android.view.Window;
11. import android.widget.Toast;
12.
13. public class MainActivity extends Activity
14. {
15. }
```

```

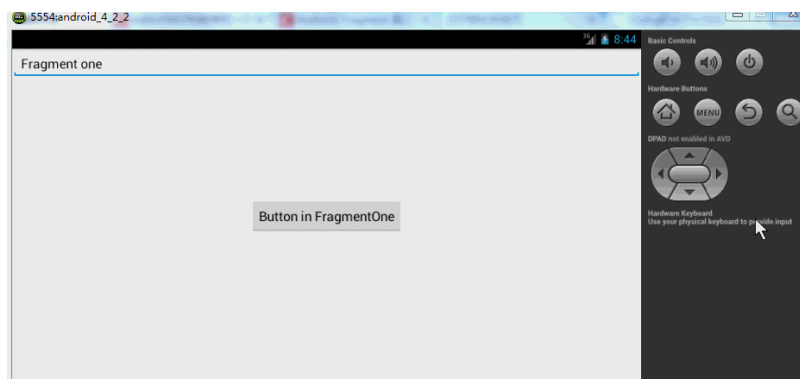
16.     private static final String TAG = "FragmentOne";
17.     private FragmentOne mFOne;
18.
19.     @Override
20.     protected void onCreate(Bundle savedInstanceState)
21.     {
22.         super.onCreate(savedInstanceState);
23.         requestWindowFeature(Window.FEATURE_NO_TITLE);
24.         setContentView(R.layout.activity_main);
25.
26.         Log.e(TAG, savedInstanceState + "");
27.
28.         if (savedInstanceState == null)
29.         {
30.             mFOne = new FragmentOne();
31.             FragmentManager fm = getFragmentManager();
32.             FragmentTransaction tx = fm.beginTransaction();
33.             tx.add(R.id.id_content, mFOne, "ONE");
34.             tx.commit();
35.         }
36.
37.     }
38.
39.     @Override
40.     public boolean onCreateOptionsMenu(Menu menu)
41.     {
42.         super.onCreateOptionsMenu(menu);
43.         getMenuInflater().inflate(R.menu.main, menu);
44.         return true;
45.     }
46.
47.     @Override
48.     public boolean onOptionsItemSelected(MenuItem item)
49.     {
50.         switch (item.getItemId())
51.         {
52.             case R.id.action_settings:
53.                 Toast.makeText(this, "setting", Toast.LENGTH_SHORT).show();
54.                 return true;
55.             default:
56.                 //如果希望Fragment自己处理MenuItem点击事件，一定不要忘了调用super.xxx
57.                 return super.onOptionsItemSelected(item);
58.         }
59.     }
60.
61. }

```

微信关注我的公众号



效果图：



好了，可以很好的看到，Fragment可以添加MenuItem，也可以自己处理点击~~~

## 6、没有布局的Fragment的作用

没有布局文件Fragment实际上是为了保存，当Activity重启时，保存大量数据准备的

请参考博客：[Android 屏幕旋转 处理 AsyncTask 和 ProgressDialog 的最佳方案](#)

## 7、使用Fragment创建对话框

这是Google推荐的方式，我也单独写过博客介绍，请参考：[Android 官方推荐：DialogFragment 创建对话框](#)

好了，终于把Fragment相关的联系到一起了，上述基本包含了Fragment所有的用法~~~相信大家如果能够看完，一定

顶

300

踩

7

我的同类文章

【Android 源码解析】（ 28 ）			【android 进阶之路】（ 64 ）		
• Volley 图片加载相关源码解析	2015-08-17	阅读 22586	• Android DiskLruCa		
• Android IntentService完全解析 当Ser...	2015-07-30	阅读 24572	• Android HandlerTh		
• Android 基于Message的进程间通信 ...	2015-07-23	阅读 27641	• Android 增强版百分比布局库 为了适...	2015-07-06	
• Android 百分比布局库(percent-supp...	2015-06-30	阅读 76262	• Android base-adapter-helper 源码分...	2015-03-02	
• Android Drawable 那些不为人知的高...	2015-02-26	阅读 85450	• Android 自定义控件 优雅实现元素间...	2015-01-08	
• Android 属性动画 源码解析 深入了解...	2014-12-25	阅读 15509			

微信关注我的公众号



参考知识库



Android知识库  
12674 关注 | 1190 收录



Docker知识库  
2363 关注 | 187 收录

猜你在找

【Android APP开发】Android高…  
Android App性能调优、内存泄露…  
Android APP开发之真机调试环境…  
android 数据解析 JSON 开发详解  
Android开发一屏幕适配

Android Fragment 真正的完全解…  
Android Fragment 真正的完全解…  
Android Fragment 真正的完全解…  
Android Fragment 真正的完全解…  
Android Fragment 真正的完全解…

阿里云 云服务器 一键购买

只需美金\$8.00 /月，享用1核 CPU，1GB内存，40GB高效云盘和2TB数据流量



○ ○

查看评论



吧主-码农出身

1

感谢楼主 无私地分享 我的 QQ: 1733563441 有一起学习的小伙伴可以 加我 楼主加我 或是我加你们  
大家一起 相互学习进步



吧主-码农出身

1


感谢楼主 无私地分享 我的 QQ: 1733563441 有一起学习的小伙伴可以 加我 楼主加我 或是我加你们  
大家一起 相互学习进步



栗夯\_

1

每天进步一点点

zzj\_jie


172楼 2016-09-01 14:28

第二个页面不是hide了吗？当从页面三返回时不需要show第二个页面吗？

Dreaming\_My\_Dream


171楼 2016-09-01 14:28

感谢哥的分亨，学到了许多东西。

C3程


170楼 2016-09-01 14:28

多谢大神分享，虽然看得有点晕，但是真的很厉害。另外，弱弱的问一句，在同一个activity中要实现startactivityForResult的效果怎么破

青圆子


169楼 2016-09-01 14:28


博主，不知你有没有遇到这个问题，当activity重启时，之前hide的fragment会和当前的fragment重叠在一起，如果是少量的fragment可以在判断savedInstanceState不为null里重新做hide和show的操作，但如果是很多怎么办呢？

晚睡晚起的

回复青圆子：你解决了吗

微信关注我的公众号



KevinMite


168楼 2016-09-01 14:28

如果想深入点了解fragment的调度机制，生命周期等，可以看下这篇博客<http://farmercoding.win/2016/07/17/fragment/>

shuai\_f

167楼 2016-09-01 14:28

牛人必须顶，已经关注

七月流觞


166楼 2016-09-01 14:28

继承了FragmentActivity，重写了onKeyDown()方法，改为按两下退出应用。  
之后FragmentOne再去addToBackStack（null），FragmentTwo不添加addToBackStack（null），从FragmentTwo回到FragmentOne的时候，back键没作用，还是执行我按两下退出的方法，这是什么冲突？？  
新人求教，希望大神解答！！

xiaoxiao701111


165楼 2016-09-01 14:28

真心的不错。

jiewin


164楼 2016-09-01 14:28

鸿洋大神可以开个培训班了，想张孝祥那样，很流弊

u014026084


163楼 2016-09-01 14:28

谢谢讲解，虽然现在还有一些不明白，但是我相信以后还会再来看这篇文章的。

干钓鱼


162楼 2016-09-01 14:28

运行时状态变化，不仅是FragmentOne会显示出来，FragmentTwo也会显示出来，而且FragmentTwo中的按键也会不起作用，好像mFTwo==null了。有没有什么简单易行的解决办法？

wang232024

161楼 2016-09-01 14:28

请问下大神，我在每个fragment的布局中添加EditView后，写入数据，再横屏的时候，数据会丢失，请问这个怎么解决呢？

readmemo

Re: 2016-09-01 14:28

回复wang232024：加一个id即可，fragment的onsaveinstance会保存数据。

lanziEVOL

160楼 2016-09-01 14:28

看了秒懂的人真是牛叉，我看的头大，晕头转向的



yylong0708

159楼 20

堪称经典，膜拜鸿洋大神！



henry\_chen\_1994

158楼 20

博主男神我爱你



xiangwang2015

157楼 20

谢谢了



WJC2015

看博主的文章，真是拨开乌云见明月，解开了我好多疑问，谢谢！



从菜鸟开始努力

大神，求源码。。280695381@qq.com



baidu\_32045567

博主写得很好，赞赞，小白学习了

微信关注我的公众号



w635254785

153楼 20

楼主我想问个问题 从fragment跳转到activity，然后在退出activity时想让fragment自动刷新其中的listview，在onResume等生命周期里都试了不好使该怎么办



Garlic\_You\_Ruthless

Re: 20

回复w635254785：首先你要明白这种界面跳转始终是Act跳Act,所以你完全可以在前者复写ActResult，然后返回的时候在ActResult里通过条件进行act控制相关的Fragment刷新就可以了！还有就是不要再生命周期函数中做逻辑处理，当然，很特别的需求除外！



啊是龙

Re: 20

回复w635254785：在退出activity的onDestroy里面刷新fragment数据



qq\_31549637

152楼 20

很好，秒懂



北国SworD

151楼 20

写的很棒



I\_IS\_FISH

150楼 20

非常感谢博主分享!!!!!!1



fzhshhxq

149楼 20

谢谢楼主这么用心的写文章！~



种子选手1021

148楼 20


准备细品，鸿洋老师的心血之作



qq\_30317573

147楼 20

qin 可以把 源代码 分享一下不，nmx2678@163.com,万分感谢

v03387


146楼 20

你好 看了你的文章相信你对Fragment很了解，想咨询一下我能在它的下面在写一个Activity文件么，初学者小白。就是滑动做好了，还做好了一个电话本能将它俩整合么。跪求大神

Xiaoxiao


145楼 20

学习了 心中只有景仰之情

Urmytch


144楼 20

第一次读博主的文章，受教了，讲解的很全面，思路清晰。比看书和文档清晰多了。向有分享精神的技术大牛致敬！以后会常来学习


Urmytch

143楼 20

第一次读博主的文章，受教了，讲解的很全面，思路清晰。比看书和文档清晰多了。向有分享精神的技术大牛致敬！以后会常来学习

super枫心

感谢

Android开始之路

好厉害，虽然刚开始学android的我不怎么看到懂，博主你有木有写什么书？或者推荐一下？视频？

小圣\_fighting

140楼 20

谢谢，学习了

AABBCC123AABBCC

139楼 20

好文章!!!!

jsryin


38楼 20

学习了，讲解的很好

无知的疯子


137楼 20

雁过留毛。两篇都看了，很有收获。谢谢。

lyabc123456


136楼 20

经过验证，第一个fragment里面的edittext如果指定了id，则返回来的时候值是可以保存的，如果editext没有指定id则不会保存

AndroidMaLin


135楼 20

这里很明显，我们调用tx.addToBackStack(null);将当前的事务添加到了回退栈，所以FragmentOne实例不会被销毁，但是视图层次依然会被销毁，即会调用onDestoryView（这个单词错了）和onCreateView（这个去掉），这段话应该修改成：  
这里很明显，我们调用tx.addToBackStack(null);将当前的事务添加到了回退栈，所以FragmentOne实例不会被销毁，但是视图层次依然会被销毁，即会调用onDestroyView

lyabc123456

134楼 20

tx.addToBackStack(null); 这句代码是将当前所在的Fragment添加到回退栈还是将上一句tx.add的Fragment添加到回退栈？有点疑惑。还有为什么这里要传个null， FragmentTransaction.addToBackStack(String)这里如果传个String参数的话需要传什么，有什么作用呢？

Ro

Re: 20

回复lyabc123456： addToBackStack是把当前事务(状态)添加到回退栈中,参数就是给这个状态取的名字,随意设置都可以

amw3323123

Re: 20

回复lyabc123456：同问



ail6413

133楼 2016-09-01 14:00

我现在在 **activity**里面使用 **viewpager+FragmentManager** 来实现多个界面的滑动，但是当屏幕发生旋转，**Activity**发生重新启动，默认的**Activity**中的**Fragment**也会跟着**Activity**重新创建，因为我这边是 使用**List<Fragment>** 来实现适配器的数据，请问这种情况 我该如何处理？ 这时候不能使用文章里面的方法



未眠的Captain

132楼 2016-09-01 13:59

博主，你好！为什么我写的**Fragment**的程序，回退栈无法使用，按下**Back**键之后，就直接退出了



WJC2015

回复未眠的Captain：如果**Fragment**是app下的，继承**Activity**有回退的效果。如果**Fragment**是V4包的，继承**Activity**没效果，你改成**FragmentActivity**试试。

微信关注我的公众号



不喝牛奶只喝酒

想问下您，我想做成网易新闻那样，上面有很多分类可滑动可点击，然后下面的**fragment**布局都一样（一个**listview**），所以我用同一个**Fragment**去new对象。为什么有时候数据加载不出来，一定要手动刷新才能出来。我是在**Fragment**创建的时候今天post请求，然后在**handler**里面处理数据和设置或刷新适配器。



不喝牛奶只喝酒

130楼 2016-09-01 13:58

想问下您，我想做成网易新闻那样，上面有很多分类可滑动可点击，然后下面的**fragment**布局都一样（一个**listview**），所以我用同一个**Fragment**去new对象。为什么有时候数据加载不出来，一定要手动刷新才能出来。我是在**Fragment**创建的时候今天post请求，然后在**handler**里面处理数据和设置或刷新适配器。



空明俗人

129楼 2016-09-01 13:57

好久没做**android**，这几天又得重新拾起来了



ydh7611

128楼 2016-09-01 13:56

这样的文章看着舒服，可以做为经典教材了。



就是爱

127楼 2016-09-01 13:55

超赞，已投票！



就是爱

126楼 2016-09-01 13:54

博客内容充实，受益匪浅，必须投票！



zst1303939801

125楼 2016-09-01 13:53

受教了



Like\_program\_aguang

124楼 2016-09-01 13:52

太牛了，膜拜大神，初学者虽然不能完全看懂，对**activity**与**fragment**的通信不是很了解。要慢慢喝博主的分享，学习到了很多，特别是最佳实践部分



leobert\_lan

123楼 2016-09-01 13:51

我疑惑的是，使用**fragment**的优点在哪里



Allen\_James

122楼 2016-09-01 13:50

鸿神，能不能给个第三个知识**fragment**与**activity**通讯的效果图？



凉城、薄梦

121楼 2016-09-01 13:49


我的和38楼问题一样 运行后按返回直接退出**activity**了 - -怎么做



NELA

120楼 2016-09-01 14:00

讲的太好了。。。。太全面了

loader

119楼 2016-09-01 13:59

引用“dywailly”的评论：


鸿洋大神，我遇到一个问题，我根据您的介绍写了一个管理fragment回退栈的代码，也是3个fragment...

鸿洋大神，我遇到一个问题，我根据您的介绍写了一个管理fragment回退栈的代码，也是3个fragment，代码基本是一样的。唯一不同的是，我一开始的时候给第一个fragment里面的edittext设置了id属性，然后回退到第一个fragment的时候，edittext里面有值。然后我把第一个fragment里面的edittext的id属性给删除了，然后再进行同样的操作,回退到frag1的时候，edittext里面没有值。请问鸿洋大神，这个是因为什么啊？？？



陌秋...

受教了。谢谢楼主，思路很清晰

sxdx\_cabbage

博主厉害！章节条理清晰，重点突出，例子的代码干净。

一休哥aaa

116楼 2016-09-01 13:58

看这样的文章真是舒服，博主大赞

曾淘


115楼 2016-09-01 13:57

不错，很清晰

yadong0724

14楼 2016-09-01 13:56

可不可以不要这么6啊

yadong0724

113楼 2016-09-01 13:55

才开始学的安卓，fragment讲的很不错。

ganoyffnm

112楼 2016-09-01 13:54

能贴个源码链接不

Mirhunana

111楼 2016-09-01 13:53

好文～

aderan

110楼 2016-09-01 13:52

谢谢分享

尧石


109楼 2016-09-01 13:51

赞一个，获益匪浅！

清澈@Cherry

108楼 2016-09-01 13:50

获得了很多启发，谢谢博主

为什么这么多叫solo


107楼 2016-09-01 13:49

学习博主的文章有一段时间了，进步还是挺明显的，现在看你写的其他博客已经不吃力了

 qczg\_wxg

楼主万岁

106楼 20

 qunidayedebi


为什么 我粘贴的到主页面的这句话tx.add(R.id.id\_content, new FragmentOne(),"ONE"); 中的add出现错误,错误是The method add(int, Fragment, String) in the type FragmentTransaction is not applicable for the arguments (int, FragmentOne, String)应该修改什么？

105楼 20

 qq\_21190705

Re: 20

回复qunidayedebi: 你看下你是不是导包导错了，一个是有support.v4的 一个是没有的

 qczg\_wxg

你确定第一种方法可以？？怎么我写的没有效果

 LJSJOE

赞一个！

 gdut\_song

很丰富很细致，要完全消化还需要实践测试测试，谢谢！

 sugaryaruan

大赞！

101楼 20

 vnking

100楼 20


不错，不错，fragment快速成就伟业。

老板陪客户喝酒，我开始写程序；  
老板陪客户上床，我写程序高潮；  
老板从床上下来，我程序已写好。

 sxyandapp

99楼 20

写的真心不错，一直对fragment不懂，现在终于懂了许多~~~

 sungangwudi


98楼 20

看了您的很多文章，真的受益匪浅，大赞！

 Z-Sunshiny

97楼 20

写的太棒了！学到好多！

 luckyzhangx


96楼 20

厉害

 suhaiqiangxue

95楼 20

好人 该赞

 LittleStone\_greensky

94楼 20

hi 鸿洋大神，fragment\_one和fragment\_two通过add加入fragmentLayout中，在fragment\_two中修改一些数据，点击回退键回到fragment\_one中，fragment\_one中的listview中数据不刷新怎么办。但是list view只要拖动下就会有数据刷新，这是什么情况？是fragment\_one中没运行生命周期么（数据都是操作的数据库）

小强\_Silence

93楼 20



学习了



挨踢老男

92楼 20

写得很好，感谢hyman的无私奉献！



dywailly

91楼 20

鸿洋大神，我遇到一个问题，我根据您的介绍写了一个管理fragment回退栈的代码，也是3个fragment，代码基本是一样的。唯一不同的是，我一开始的时候给第一个fragment里面的edittext设置了id属性，然后回退到第一个fragment的时候，edittext里面有值。然后我把第一个fragment里面的edittext的id属性给删除了，然后再进行同样的操作,回退到frag1的时候，edittext里面没有值。请问鸿洋大神，这个是因为什么啊？？？



让定律更简单

写的一手风骚的代码

微信关注我的公众号



zhongyuanceshi

博主太棒了，牛X呀



qq\_21582695

要用add的方式

88楼 20



qq\_21582695

87楼 20

额 这个怎么在一个fragment中设置点击事件然后再切换到另一个fragment中啊

查看更多评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack	VPI
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI	HTML5	Spring
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rai
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	S	
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr	An
Cloud Foundry	Redis	Scala	Django	Bootstrap							

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved