Fragment的概念

（1）Fragment是在Android（API level11）开始引入的新的API技术。

（2）为了提高代码重用性和改善用户体验，我们将Activity中的UI组件进行分组和模块化管理。这些分组后的UI组件就是Fragment。

（3）一个Activity页面中可以包含多个Fragment模块，而同一个Fragment模块也可以被多个Activity使用。

（4）每个Fragment有自己的布局，有自己的生命周期。

（5）Fragment必须放在一个Activity中

（6）Fragment有它自己的生命周期，而且受到它所在的宿主Activity的生命周期的影响。

（7）Fragment可以接收它自己的事件

（8）一个Fragment可以放在多个Activity中，一个Activity中也可以放置多个Fragments。

创建一个Fragment要实现的回调函数

onCreate()

创建Fragment的时候系统会调用此函数。

onPause()

当用户离开当前Fragment时调用这个方法。通常用来保存持久化数据

onCreateView()

当Fragment第一次绘制它的UI的时候调用。

这个方法必须返回一个View，

表示这个Fragment的根布局

当首次进入程序时：

MainActivity onCreate

MyFragment onAttach

MyFragment onCreate

MyFragment onCreateView

MyFragment onActivityCreated

MainActivity onStart

MyFragment onstart

MainActivity onResume

MyFragment onResume

当点击home键切换到桌面时：

MyFragment onPause

MainActivity onPause

MyFragment onStop

MainActivity onStop

当重新切换回Activity时

MainActivity onStart

MyFragment onstart

MainActivity onResume

MyFragment onResume

退出Activity时

MyFragment onPause

MainActivity onPause

MyFragment onStop

MainActivity onStop

MyFragment onDestroyView

MyFragment onDestroy

MyFragment onDetach

MainActivity onDestroy

Fragment的生命周期



一个Fragment可以处于3种状态：

运行状态 Fragment在当前运行的Activity中可见

暂停状态 另一个Activity在前台，具有焦点，但是Fragment所在的Activity依然可见

停止状态 Fragment不可见。宿主Activity处于停止状态或者Fragment已经从Activity删除并添加到后台栈中。

Fragment的生命周期

Fragment的宿主Activity的生命周期直接影响Fragment的生命周期。因此，Activity的每个生命周期回调函数都会调用对应的Fragment的回调函数。例如，当宿主Activity的onPause()被调用时，Fragment的onPause()也会被调用。

Fragment还有一些其他的回调函数与宿主Activity交互，处理UI的创建与销毁。

onAttach() 当Fragment与Activity建立关联的时候调用，Activity作为参数传入。

当Activity的onCreate()方法调用完成时返回执行

onCreateView() 创建Fragment的UI

onActivityCreated() 当Activity创建完成后回调此方法

onDestroyView() 当Fragment的UI被删除的时候调用

onDetach() 当Fragment与Activity取消关联的时候调用

当宿主Activity处于运行状态可以自由的添加和删除Fragment。只有当Activity处于运行状态的时候，Fragment的生命周期才能独立变化。否则，直接受宿主Activity生命周期的影响。

fragment\_demo.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"**>  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"** />  
</**LinearLayout**>

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**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"**>  
  
 *<!--静态放置一个Fragment-->* <**fragment  
 android:id="@+id/myfragment1"  
 android:name="com.amaker.hjs.launchmodedemo.fragment.MyFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>  
  
  
</**RelativeLayout**>

MyFragment.java

**package** com.amaker.hjs.launchmodedemo.fragment;  
  
**import** android.arch.lifecycle.Lifecycle;  
**import** android.content.Context;  
**import** android.os.Bundle;  
**import** android.support.annotation.NonNull;  
**import** android.support.annotation.Nullable;  
**import** android.support.v4.app.Fragment;  
**import** android.util.Log;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
  
**public class** MyFragment **extends** Fragment {  
 **private static final** String ***TAG***=**"MyFragment"**;  
  
  
 */\*\*  
 \* Activity的onCreate()方法调用完成之后，回调此方法  
 \** ***@param context*** *\*/* @Override  
 **public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 Log.*i*(***TAG***,**"onAttach"**);  
 }  
  
  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 Log.*i*(***TAG***,**"onCreate"**);  
 }  
  
  
  
 */\*\*  
 \* 显示的视图  
 \** ***@param inflater*** *基于布局的解析  
 \** ***@param container*** *\** ***@param savedInstanceState*** *\** ***@return*** *\*/* @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 Log.*i*(***TAG***,**"onCreateView"**);  
 View view = inflater.inflate(R.layout.***fragment\_demo***,**null**);  
 **return** view;  
 }  
  
  
  
 */\*\*  
 \* Activity已经创建起来  
 \** ***@param savedInstanceState*** *\*/* @Override  
 **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  
 **super**.onActivityCreated(savedInstanceState);  
 Log.*i*(***TAG***,**"onActivityCreated"**);  
 }  
  
  
 @Override  
 **public void** onStart() {  
 **super**.onStart();  
 Log.*i*(***TAG***,**"onStart"**);  
 }  
  
  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 Log.*i*(***TAG***,**"onResume"**);  
 }  
  
  
 @Override  
 **public void** onPause() {  
 **super**.onPause();  
 Log.*i*(***TAG***,**"onPause"**);  
 }  
  
  
 @Override  
 **public void** onStop() {  
 **super**.onStop();  
 Log.*i*(***TAG***,**"onStop"**);  
 }  
  
 @Override  
 **public void** onDestroy() {  
 **super**.onDestroy();  
 Log.*i*(***TAG***,**"onDestroy"**);  
 }  
  
  
 */\*\*  
 \* 销毁视图  
 \*/* @Override  
 **public void** onDestroyView() {  
 **super**.onDestroyView();  
 Log.*i*(***TAG***,**"onDestroyView"**);  
 }  
  
  
 */\*\*  
 \* 当前Fragment与Activity取消关联  
 \*/* @Override  
 **public void** onDetach() {  
 **super**.onDetach();  
 Log.*i*(***TAG***,**"onDetach"**);  
 }  
}

MainActivity.java

**package** com.amaker.hjs.launchmodedemo.fragment;  
  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private static final** String ***TAG*** = **"MainActivity"**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 Log.*i*(***TAG***,**"onCreate"**);  
 }  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
 Log.*i*(***TAG***,**"onStart"**);  
 }  
  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
 Log.*i*(***TAG***,**"onResume"**);  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 Log.*i*(***TAG***,**"onPause"**);  
 }  
  
 @Override  
 **protected void** onStop() {  
 **super**.onStop();  
 Log.*i*(***TAG***,**"onStop"**);  
 }  
  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 Log.*i*(***TAG***,**"onDestroy"**);  
 }  
}