

ViewPager: 安卓3.0以上可以使用的控件, 功能就是使视图左右滑动. 在`android.support.v4.view`包下, 是个扩展的自定义控件.

故使格式: 在布局文件里应该

是`android.support.v4.view.ViewPager`

注意ViewPager的内部pager页面不能使用Activity与Fragment原因:

不能使用activity的原因: viewpager中存放的是view控件, activity不能返回view, activity是不能在viewpager中使用

不能使用Fragment的原因: (以下是特有现象, 但如果不是这种fragment多次嵌套fragment的话, 可以使用fragment)

因为之前已经使用了HomeFragment, 在接下来的操作还会有新界面创建,

这样就会出现一个Fragment中使用了另一fragment, 另一个fragment中又使用了一个fragment的现象, 对于小内存的手机来说, 容易内存溢出(OOM)

创建流程: (注意: 这个组件是用来显示左右滑动的界面的, 如果不加载xml布局文件, 他是不会显示内容的。)

其创建的大致流程, 写法和ListView差不多. 但在ViewPager适配器里的`instantiateItem`方法有很大不同.

1.在所要使用的Activity的布局xml文件里定义这个组件，

```
<android.support.v4.view.ViewPager  
    android:id="@+id/viewpager"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

提示:**android:clipChildren="false"** **ViewPager**加上这个属性后,就可以让中间的**item**变大显示

(参考网

址:<http://blog.csdn.net/flymoon1201/article/details/44646473>)

2.在Activity里找到ViewPager控件,并得到ViewPager控件对象.

```
ViewPager viewPager = (ViewPager)  
findViewById(R.id.viewPager);
```

3.准备数据,创建ArrayList集合,在创建一个承载数据类型的bean类,把数据添加到bean类中,再把bean类挨个添加到集合中.

```
ArrayList<bean> arrayList=new ArrayList<bean>  
( );
```

4.为ViewPager设置Adapter适配器

```
viewPager.setAdapter(new 自定义的适配器类());
```

5. 创建一个自定义的ViewPager适配器内部类, 要继承PagerAdapter, 并复写其方法

```
class MyAdapter extends PagerAdapter{  
    //设置ViewPager的条目个数.  
    public int getCount() {  
//设置数小的话, 图片到头了就没法循环出现, 要想循环播  
放, 一般指定Integer.MAX_VALUE次即  
可. return Integer.MAX_VALUE;  
//且报空指针异常的地方(一般和获取ViewPager当前选中  
的是第几页有关的都要改为position(当前页  
数)%list.size()(取余数)  
        return list.size();  
    }  
}
```

提示: 如果想要ViewPager能向左滑动, 只需要在主线程, 改变ViewPager的默认选中位置即

可: viewPager.setCurrentItem(int);

```
    //判断ViewPager的条目View对象和  
InstantiateItem返回的Object对象是否一致, 固定格  
式: return view==object;  
    public boolean isViewFromObject(View  
view, Object object) {  
        return view == object; //这里是固定写  
法, 一般不用变, 写死了.
```

```
}
```

```
/**
```

* 创建Pager的方法，相当于BaseAdapter的getView方法,很重要.(就是没有position类似的参数,所以没有优化)

* 一般在该方法中要加载并返回View对象，并且要绑定数据,但是不需要复用Pager了,且返回container没有用.

```
*/
```

//注意:是参数为ViewGroup container(container代表了ViewPager控件自身)的instantiateItem方法第二个,不要弄错了

```
public Object instantiateItem(ViewGroup  
container, int position) {
```

```
    //1.加载View
```

```
    ImageView imageView = new  
    ImageView(MainActivity.this);
```

```
    imageView.setScaleType(ScaleType.FIT_XY); //设置  
    所包裹的图片占满整个屏幕
```

```
    //2.绑定数据到View
```

```
    imageView.setImageResource(list.get(position).g  
    etIconId());
```

```
    //注意：必须要将控件对象添加到
```

ViewPager中来

```
        container.addView(imageView);  
        return imageView;  
    }  
    /**防止内存泄漏.相当于ListView的复用
```

container

* 销毁一个page,该方法的实际就是将
instantiateItem返回的View对象从ViewPager中移除,
container:还是ViewPager控件自
身,object:则代表了View控件,使用时要强转成View一下
补充:为什么参数是Object,而不直接是
View,因为虽然99%是view,但也有可能是Fragment,所以
用Object,提高了可扩展性.

```
    */  
    public void destroyItem(ViewGroup  
container, int position, Object object) {  
    //          super.destroyItem(container,  
position, object);此方法一定要删掉.  
        container.removeView((View)  
object);  
    }
```

6.为ViewPager设置页面切换监听器.并在
onPageSelected里进行文本的设置(所有的 position :
条目的位置)

viewPager.addOnPageChangeListener(new
OnPageChangeListener() { //也可以是
setOnPageChangeListener, 二者是一样的, 只不过set被
淘汰了

// ViewPager的页面pager切换完成时, 执行
该方法

```
        public void onPageSelected(int  
position) {  
            Log.e("log", "当页面改变时执  
行:onPageSelected");  
            updateTitleAndDot() //记得在第4步, 为ViewPager  
设置Adapter适配器下面, 调用该方法, 使ViewPager一创  
建, 第一个页面就有文本数据  
        }
```

//ViewPager控件处于滑动中, 执行该方法, 参数信息
positionOffset: 页面移动距离的百分比例
positionOffsetPixels : 移动的偏移的像素
//注意: positionOffsetPixels, positionOffset: 会随
着页面向右滑动而不断增加(无限接近于1), 但当完全滑动
到另一页时, 数据就变成了0.0

```
        public void onPageScrolled(int  
position, float positionOffset, int  
positionOffsetPixels) {  
            Log.d("log", "当页面处于滑动中时执
```

```
行:onPageScrolled");  
    }
```

// 当ViewPager控件被点击或松开时(也可以理解为pager滑动状态改变时),执行该方法

```
    public void  
onPageScrollStateChanged(int state) {  
        Log.i("log","当页面时执  
行:onPageScrollstateChanged");  
    }  
}  
);
```

7.怎么通过做出一个简单的图形?显示一排小圆点,在主线程准备完数据后,就要使用该方法显示出来.

```
    private void initDot() {  
        for(int i=0;i<list.size();i++){  
            //创建点  
            View view = new View(this);  
            //给view设置背景图片,就是黑点的图片  
  
view.setBackgroundResource(R.drawable.dot_focus  
);  
  
            //设置属性的宽高  
            LayoutParams params = new
```

```
LayoutParams(8, 8);
        params.leftMargin = 8;
//将view对象添加到ll_dot(一个LinearLayout)布局中,
之所以用布局控件,是为了可以动态的添加点的个数,把他
当做一个容器.
        ll_dot.addView(view, params);//把
View添加到容器中,并把设置的属性对象也添加进去.
    }
}
```

8. 根据当前选中的页(getCurrentItem()是ViewPager的方法,可以直接拿到正在选中的ViewPager所在的位置),让对应的标题和点进行改变状态:

```
private void updateTitleAndDot(){
    //获取ViewPager当前选中的的是第几页,返回结果
是一个Int型
    int currentItem =
viewPager.getCurrentItem();
    //让tv_title设置指定位置的标题

tv_title.setText(list.get(currentItem).getTitle
());
    //根据当前是第几页,让第几个点设置白色点
    for (int i = 0; i < list.size(); i++) {
        View child =
```


`ll_dot.getChildAt(i);`//获取调用控件的int指定位置的子控件.

```
child.setBackgroundResource(i==currentItem?  
R.drawable.dot_focus:R.drawable.dot_unfocus);  
    }  
}
```

9.使用handler定时循环自动播放ViewPager页面(注意:使用handler定时循环会导致内存泄漏,所以一定要有移除handler循环的代码,比如在Activity的OnDestroy方法进行移除.)

```
private Handler handler = new Handler(){  
    public void handleMessage(Message msg)  
{  
        super.handleMessage(msg);  
        //          Log.e("tag", "handleMessage");  
        //收到消息,先选中下一页(setCurrentItemViewPager  
        的方法,可以跳转到指定页,一般和getCurrentItem一起使  
        用),然后再次发送延时消息.
```

```
viewPager.setCurrentItem(viewPager.getCurrentItem()  
+1);//viewPager.getCurrentItem()获取当前  
页,+1就是下一页了.
```

```
handler.sendMessageDelayed(9(what), 3000(时间));  
  
    }  
  
};
```

提示:在主线程发送空的消息

```
handler.sendMessageDelayed(9, 3000);
```

`sendMessage(int)`,空消息是指该消息仅包含what值。

`sendMessageDelayed(int what, long delayMillis)`,在指定的时间后发送空消息,指定的时间以ms为单位。

`sendMessageAtTime(int what, long uptimeMillis)`,在指定时间点发送空消息, `uptimeMillis`是指本次开机开始运行的时间点,不包含系统休眠的时间,单位为ms。

10.当点击条目时,自动停止使用handler定时循环播放ViewPager页面,当不点击时,恢复使用handler定时循环播

放ViewPager页面。

```
vp.setOnTouchListener(new OnTouchListener()
{
    public boolean onTouch(View v,
MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:

handler.removeCallbacksAndMessages(null);
                break;
            case MotionEvent.ACTION_MOVE:

handler.removeCallbacksAndMessages(null);
                break;
            case MotionEvent.ACTION_UP:

handler.sendEmptyMessageDelayed(0, 2000);
                break;
            default:
                return true;
        }
        return false;
    }
});
```

ViewPager对象可以调用的各种方法:

ViewPager对象.setCurrentItem(int);//使ViewPager跳转到指定的页面(就是设置ViewPager当前显示的界面).
返回结果:void

ViewPager对象.getCurrentItem(int);//获取到ViewPager的当前页面位置,返回结果:int Positiono