转载请注明链接：https://blog.csdn.net/feather_wch/article/details/79494748

> 讲解ViewPager的使用方法，包括ViewPager做启动页面(加载一般的View)，以及ViewPager+Fragment实现多个Fragment左右滑动切换的功能。

ViewPager使用方法

版本：2018/9/5-1

ViewPager

1、ViewPager是什么?

> 1. ViewPager是 `android扩展包v4包` 中的类，这个类可以让我们左右切换当前的view。

2、ViewPager的特点

> 1. `ViewPager类` 直接继承了 `ViewGroup类` ，因此它一个容器类，可以添加其他的view类
> 2. `ViewPager类` 需要一个 `PagerAdapter` 适配器类给它提供数据（类似于 `ListView` 所需要数据适配器Adater）
> 3. `ViewPager` 经常和Fragment一起使用，并且官方还提供了专门的 `FragmentPagerAdapter和FragmentStatePagerAdapter` 类供Fragment中的ViewPager使用

3、ViewPager启动页面的实现方法

> 1. 自定义多个启动页面的布局-layout_tab1,layout_tab2,layout_tab3等

```
//layout_tab1
<?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"
    android:background="@color/colorAccent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="tab1"/>
</LinearLayout>
//layout_tab2等类似
......
```

> 2. Activity的布局

```
//activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.a6005001819.androiddeveloper.MainActivity">

    <android.support.v4.view.ViewPager
        android:id="@+id/main_viewpager"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </android.support.v4.view.ViewPager>

</android.support.constraint.ConstraintLayout>
```

> 3. 继承并实现 `PagerAdapter` 中的方法，并且在 `Activity` 中使用

```java
//MainActivity.java
package com.example.a6005001819.androiddeveloper;

import android.app.Activity;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;
import android.os.Handler;
import android.support.v4.view.PagerAdapter;
import android.support.v4.view.ViewPager;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewParent;

import java.util.ArrayList;

public class MainJavaActivity extends Activity implements BlankFragment.onFragmentInteractionLi

    ArrayList<View> viewContainter = new ArrayList<View>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_java);
        //1. 将View视图都存入列表中
        View view1 = LayoutInflater.from(this).inflate(R.layout.layout_tab1, null);
        View view2 = LayoutInflater.from(this).inflate(R.layout.layout_tab2, null);
        View view3 = LayoutInflater.from(this).inflate(R.layout.layout_tab3, null);
        viewContainter.add(view1);
        viewContainter.add(view2);
        viewContainter.add(view3);
        //2. 给ViewPager设置适配器
        ViewPager viewPager = findViewById(R.id.main_viewpager);
        viewPager.setAdapter(new MyPagerAdapter());
    }

    /**
     * 3. 自定义适配器，需要实现下面的"四个方法"
     */
    public class MyPagerAdapter extends PagerAdapter {
        //1. 返回可以滑动的View的个数
        @Override
        public int getCount() {
            return viewContainter.size();
        }
        //2. 滑动切换时销毁当前View
        @Override
        public void destroyItem(ViewGroup container, int position, Object object) {
            container.removeView(viewContainter.get(position));
        }
        //3. 将当前View添加到父容器中，并且返回当前View
        @Override
```

```
        public Object instantiateItem(ViewGroup container, int position) {
            container.addView(viewContainter.get(position));
            return viewContainter.get(position);
        }
        //4. 确认"第三步"instantiateItem返回的Object与页面View是否是同一个
        @Override
        public boolean isViewFromObject(View view, Object object) {
            //5. 官方推荐直接 `view == object`
            return view == object;
        }
    }
}
```

4、PagerTitleStrip类和PagerTabStrip类(不推荐使用)

1. 作用：用于设置每页的标题
2. 缺点： 标题会随着页面滑动，建议使用自定义的指示器

ViewPager+Fragment

5、FragmentPagerAdapter与FragmentStatePagerAdapter

1. 用于 Fragment 需要两个特殊的适配器 FragmentPagerAdapter 与 FragmentStatePagerAdapter
2. FragmentPagerAdapter 会将每个生成的 Fragment 都保存在内存中，适合 相对静态，数量较少 的情况
3. FragmentStatePagerAdapter 适合 数据动态，页面数量较多，占用内存多 的情况

6、FragmentPagerAdapter与FragmentStatePagerAdapter的使用

1. 直接继承不同Adpater即可，内部代码完全一致。
2. FragmentStatePagerAdapter 会根据 Fragment状态 去销毁不是当前页面的Fragment，从而节约内存。

```
 public class MyFragmentPagerAdapter extends FragmentPagerAdapter{
    List<Fragment> fragmentList;
    public MyFragmentPagerAdapter(android.support.v4.app.FragmentManager fm, List<Fragment>
        super(fm);
        fragmentList = list;
    }
    @Override
    public Fragment getItem(int position) {
        return fragmentList.get(position);
    }
    @Override
    public int getCount() {
        return fragmentList.size();
    }
}
public class MyFragmentStatePagerAdapter extends FragmentStatePagerAdapter{
    List<Fragment> fragmentList;
    public MyFragmentStatePagerAdapter(android.support.v4.app.FragmentManager fm, List<Frag
        super(fm);
        fragmentList = list;
    }
    @Override
    public Fragment getItem(int position) {
        return fragmentList.get(position);
    }
    @Override
    public int getCount() {
        return fragmentList.size();
    }
}
```

7、实现ViewPager+Fragment的滑动

> 1. 实现需要加载的各种 `Fragment` ， 实现方法不赘述
> 2. `Activity` 中进行加载，且该Activity 必须继承自`FragmentActivity`

```
//MainActivity.java
//1、新建Fragment存入链表
ArrayList<Fragment> list = new ArrayList<>();
list.add(new Fragment1());
list.add(new Fragment2());
//2、ViewPager设置适配器
ViewPager viewPager = findViewById(R.id.main_viewpager);
viewPager.setAdapter(new MyFragmentStatePagerAdapter(getSupportFragmentManager(), list));
```

无限循环滑动

1、ViewPager实现无限循环滑动的方法?

> 1. 第一种：自定义ViewPager，重写其中的某些方法，比如滑到最后一页后，从第一页开始滑动。缺点工作量大，会卡顿。
> 2. 第二种：在Adapter中对数据进行处理，实现无限循环。

参考资料

1. ViewPager系列教学