1/20-1/22 编程界面打包

参考链接：<https://www.imooc.com/video/2213>

题目：有结果返回的页面翻转

这一章我经历的时间有点长，主要是一个容易被遗漏的bug，我希望能完成第二个页面到第三个页面的跳转—当我点击listView中的单元时，会进入第三个页面进行条件或者动作的选择，选择完再返回到第二个页面，并且更新第二个页面。

首先我们创建TActivity并且让它继承Activity

**public class** TActivity **extends** Activity

然后创建tactivity的constraint layout布局，创建中会遇到的问题详见第一章，一下代码是tactivity中布置的测试代码

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**Button  
 android:id="@+id/choice1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"  
 android:layout\_marginRight="8dp"  
 app:layout\_constraintRight\_toRightOf="parent"  
 android:layout\_marginLeft="8dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:layout\_marginTop="8dp"** />  
</**android.support.constraint.ConstraintLayout**>

接下来我们来填写TActivity的内容：

1. 引入布局

setContentView(R.layout.***tactivity***);

1. 声明和初始化按钮

**public class** TActivity **extends** Activity{  
 Button **choice1**;  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 *//\*\*\*\*\*\*\*\*\*\*\*\*\*重要,注册* setContentView(R.layout.***tactivity***);  
 **choice1**=(Button)findViewById(R.id.***choice1***); }  
}

1. 设置intent事件

在onCreate里对choice1进行设置setOnClickListener点击的时候就会返回第二个页面并且传递返回值，具体的步骤如下

初始化一个intent叫做data用来存返回值，这里的data里存的是键—值对(和map一样)

Intent data = **new** Intent();  
data.putExtra(**"choice"**,**"choice1"**);

这样我就可以取得data的choice里的内容了

设置页面返回的标识为2，因为待会需要在第二个页面中设置页面请求标识1和返回标识2，这样页面与页面的翻转和传值就不会弄错了。

setResult(2,data);  
finish();

当执行finish时，第三个页面的任务就完成了，翻转回第二个页面。

第三个页面总体的代码是这样的：

**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by 1535725170 on 2018/1/22.  
 \*/  
  
//1.创建并且extend Activity  
 //2.创建tactivity为constraintlayout,如果生成失败，回看之前创建这种layout时的教程  
 //3.在tactivity里创建一个按钮进行测试(声明+初始化）***public class** TActivity **extends** Activity{  
 Button **choice1**;  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 *//\*\*\*\*\*\*\*\*\*\*\*\*\*重要,注册* setContentView(R.layout.***tactivity***);  
 **choice1**=(Button)findViewById(R.id.***choice1***);  
  
 *//4.对button设置intent监听事件* **choice1**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//返回数据为data* Intent data = **new** Intent();  
 data.putExtra(**"choice"**,**"choice1"**);  
 *//设置页面返回标识为2* setResult(2,data);  
 finish();  
 }  
 });  
 }  
}

然后我们就回到SAvtivity写代码，需要改变的地方主要也和intent有关

还记得来listView时有个关于listView中单元的点击函数

**public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id)

吗，我们需要在这个函数里写入intent的内容。

如下:

**public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {

//从SActivity跳转到TActivityIntent intent = **new** Intent(SActivity.**this**,TActivity.**class**);

*//第二个参数是请求的标识（见上文中请求标识与返回标识）——请求码* startActivityForResult(intent,1);  
}

这个时候会多出来一个函数，是关于startActivityForResult之后得到返回值的操作的，也就是我们从哪得到返回值，用这些返回值做什么

如下：

**protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if**((requestCode==1) && (resultCode==2)){  
 String content = data.getStringExtra(**"choice"**);  
 **test**.setText(content);  
 }  
}

这里重要的是判断请求标识和返回标识，这样就能得到正确的返回值了。

String content = data.getStringExtra(**"choice"**);是按照前面说的得到键choice所对应的值，至于最后一句是我写的测试代码，把结果显示出来。

这样我们就可以开始跑一下了，但是发现想要跳转到第三个页面时，程序崩了，这是怎么回事呢？我这两天就是倒在这一个地方，试了n种方法最后才发现，是TActivity没有注册

非常重要：  
在manifest文件中，我们要像注册FActivity和SActivity一样注册TActivity，这样就可以用了

<**activity android:name=".TActivity"**>  
</**activity**>

这样，我们就完成了最初步的功能。这里我就不放manifest中的代码了。

SActivity中：

**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.annotation.LayoutRes;  
**import** android.support.annotation.Nullable;  
**import** android.transition.TransitionManager;  
**import** android.view.View;  
**import** android.widget.AbsListView;  
**import** android.widget.AdapterView;  
**import** android.widget.Button;  
**import** android.widget.LinearLayout;  
**import** android.widget.ListView;  
**import** android.widget.SimpleAdapter;  
**import** android.widget.TextView;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.List;  
**import** java.util.Map;  
**import** java.util.Objects;  
  
*/\*\*  
 \* Created by 1535725170 on 2018/1/15.  
 \*/  
  
//实现listview点击和滚动 OnItemClickListener,OnScrollListener 并且点击小红灯import method来实现相关接口中的方法***public class** SActivity **extends** Activity **implements** View.OnClickListener,AdapterView.OnItemClickListener,AbsListView.OnScrollListener {  
 **private** Button **to\_code\_screen**;  
 **private** Button **return\_to\_majot\_screen**;  
 **private** TextView **test**;  
  
  
  
 *//声明listview和simple\_adapter* **private** ListView **listView**;  
 **private** SimpleAdapter **simp\_adapter**;  
  
 *//声明dataList（作为适配器的第二个参数）,导入包* **private** List<Map<String,Object>>**dataList**;  
  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***sactivity***);  
 **to\_code\_screen**=(Button)findViewById(R.id.***to\_code\_screen***);  
 **return\_to\_majot\_screen**=(Button)findViewById(R.id.***return\_to\_majot\_screen***);  
 **test** = (TextView)findViewById(R.id.***test***);  
  
 *//初始化listview* **listView**=(ListView)findViewById(R.id.***listView***);  
  
 *//初始化dataList* **dataList**=**new** ArrayList<Map<String,Object>>();  
  
 *//新建数据适配器  
 //SimpleAdapter（）  
 /\*\*  
 \* context：上下文  
 \* data：数据源（List<？ extends Map<String,?>> data)一个Map所组成的List集合  
 \* 每一个Map都会对应ListView列表中的一行  
 \* 每一个Map（键-值对）中的键必须包含所有在from中所指定的键  
 \* resource：列表中布局文件的id  
 \* from:Map中的键名  
 \* to:绑定数据视图中的ID，与from成对应关系  
 \*/* **simp\_adapter** = **new** SimpleAdapter(**this**,getData(),R.layout.***item***,**new** String[]{**"pic"**,**"text"**},**new int**[]{R.id.***pic***,R.id.***code***});  
  
 *//新建item布局文件  
  
 //视图（listview）加载适配器* **listView**.setAdapter(**simp\_adapter**);  
 *//加载监听器\*2* **listView**.setOnItemClickListener(**this**);  
 **listView**.setOnScrollListener(**this**);  
  
 **to\_code\_screen**.setOnClickListener(**this**);  
 **return\_to\_majot\_screen**.setOnClickListener(**this**);  
  
  
  
 }  
  
 *//写得到数据源的函数* **private** List<Map<String,Object>> getData(){  
 *//写一个测试代码* **for**(**int** i=0;i<20;i++){  
 Map<String,Object>map = **new** HashMap<String,Object>();  
 map.put(**"pic"**,R.mipmap.***ic\_launcher\_round***);  
 map.put(**"text"**,**"huhu"**+i);  
 **dataList**.add(map);  
 }  
  
  
  
 **return dataList**;  
 }  
  
 @Override  
 **public void** onClick(View v) {  
 String output = **"ccc"**;  
 ManyTreeNode t1 = **new** ManyTreeNode(**"1"**);  
 ManyTreeNode t2 = **new** ManyTreeNode(**"2"**);  
 ManyTreeNode t3 = **new** ManyTreeNode(**"3"**);  
 ManyTreeNode t4 = **new** ManyTreeNode(**"4"**);  
 ManyTreeNode t5 = **new** ManyTreeNode(**"5"**);  
 ManyTreeNode t6 = **new** ManyTreeNode(**"6"**);  
 ManyNodeTree m = **new** ManyNodeTree();  
 m.addChild(t1);  
 m.addChild(t2);  
 m.addChild(t3);  
 m.beChild(1);  
 m.addChild(t4);  
 m.addChild(t5);  
 m.addChild(t6);  
 m.beParent();  
 m.addChild(t6);  
 m.addChild(t6);  
 m.deleteChild(2);  
  
 **switch** (v.getId()){  
 **case** R.id.***to\_code\_screen***:  
 output=m.iteratorTree(m.getRoot());  
 **break**;  
 **case** R.id.***return\_to\_majot\_screen***:  
 output=**"bbb"**;  
  
  
 **break**;  
 **default**:  
 **break**;  
 }  
 **test**.setText(output);  
 }  
  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 *//测试点击事件  
 //String textt = listView.getItemIdAtPosition(position)+" ";  
  
 //test.setText(textt);* Intent intent = **new** Intent(SActivity.**this**,TActivity.**class**);  
 *//第二个参数是请求的标识——请求码* startActivityForResult(intent,1);  
  
  
 }  
  
  
  
 *//笔记：刷新适配器: 对dataList进行一波操作之后，刷新适配器 simp\_adapter.notifyDataSetChanged()* @Override  
 **public void** onScrollStateChanged(AbsListView view, **int** scrollState) {  
 *//这里可以改变下拉时的一些功能，如下拉刷新* }  
  
 @Override  
 **public void** onScroll(AbsListView view, **int** firstVisibleItem, **int** visibleItemCount, **int** totalItemCount) {  
  
 }  
  
  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if**((requestCode==1) && (resultCode==2)){  
 String content = data.getStringExtra(**"choice"**);  
 **test**.setText(content);  
 }  
 }  
}

TActivity中：

**import** android.app.Activity;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.annotation.Nullable;  
**import** android.view.View;  
**import** android.widget.Button;  
  
*/\*\*  
 \* Created by 1535725170 on 2018/1/22.  
 \*/  
  
//1.创建并且extend Activity  
 //2.创建tactivity为constraintlayout,如果生成失败，回看之前创建这种layout时的教程  
 //3.在tactivity里创建一个按钮进行测试(声明+初始化）***public class** TActivity **extends** Activity{  
 Button **choice1**;  
 @Override  
 **protected void** onCreate(@Nullable Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
  
 *//\*\*\*\*\*\*\*\*\*\*\*\*\*重要,注册* setContentView(R.layout.***tactivity***);  
 **choice1**=(Button)findViewById(R.id.***choice1***);  
  
 *//4.对button设置intent监听事件* **choice1**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//返回数据为data* Intent data = **new** Intent();  
 data.putExtra(**"choice"**,**"choice1"**);  
 *//设置页面返回标识为2* setResult(2,data);  
 finish();  
 }  
 });  
 }  
}