

安卓的活动

安卓

这一次我们主要学习的是安卓的 Activity

• 一个完整的活动注册

```
1. //1.在activity文件中指定
2. setContentView(R.layout.dialog_layout); /*指定使用的布局文件*/
3. //2.在Androidmanifest中
4.
5. <activity android:name=".MainActivity">
6.     <intent-filter>
7.         <action android:name="android.intent.action.MAIN" />
8.         <category android:name="android.intent.category.LAUNCHER"/>
9.     </intent-filter>
10. </activity>
```

• 不同activity之间的通讯

显式传递

```
1. //将页面从当前的 FirstActivity 跳转到 SecondActivity
2. Intent intent = new Intent(FirstActivity.this, SecondActivity.class);
3. //启动
4. startActivity(intent);
```

隐式传递

```
1. Intent intent = new Intent("com.example.activity.ACTION_START");//主匹
   配
2. intent.addCategory("com.example.activitytest.MY_CATEGORY");//副匹配
3.
4. //必须在Androidmanifest设置好相应的匹配
```

• 向下一个活动传递数据

```
1.  /*****发送方（当前页面）*****/
2.  String data = "hello this is first";
3.  Intent intent = new Intent(FirstActivity.this.SecondActivity.class);
4.  intent.putExtra("key",data);
5.  startActivity(intent);
6.  /*****接受方（下一个页面）*****/
7.  Intent intent = getIntent();//获取用于启动SecondActivity的Intent
8.  String data = getIntent().getStringExtra("key");//取出数据
9.  Log.d("SecondActivity",data);
```

• 返回数据给上一个界面

```
1.  /*****当前活动*****/
2.  Intent intent = new Intent(FirstActivity.this,SecondActivity.class);
3.  startActivityForResult(intent,1);//1作为标记用来返回的时候识别是哪个intent
4.  /*****第二个活动*****/
5.  Intent intent = new Intent();
6.  intent.putExtra("key","hello this is second");
7.  setResult(RESULT_OK,intent);
8.  finish();//销毁
9.  /***第一个活动**/
10. //！！重写一下 onActivityResult()方法！！
11. switch(requestCode) {
12.     case 1 :
13.         if(resultCode==RESULT_OK) {
14.             String returnData = intent.getStringExtra("key");
15.             Log.d("FirstActivity",returnData);
16.         }
17.         default:
18.     }
```

• 活动的生命周期

活动状态

状态	说明
运行状态	位于栈顶，系统不收回
暂停状态	不处于栈顶但仍然可见（支付宝输密码的地方）系统一般不收回，除非内存很少

状态	说明
停止状态	不处于栈顶，并且不可见，不可靠，容易被回收
销毁状态	从栈移除，系统回收

活动的生存期

回调方法	解释
onCreate	活动第一次被创建的时候执行，完成初始化操作，如加载布局和绑定事件
onStart	活动由不可见变成可见的时候执行
onResume	在活动准备好和用户交互的时候执行
onPause	系统准备去启动或者回复另一个活动的时候执行
onStop	当一个活动完全不可见的时候执行
onDestroy	在活动被销毁之前执行
onRestart	活动重新启动的时候执行

● 解决活动回收导致的数据丢失问题

`onSaveInstanceState` 方法

```

1.  /*****将要被结束的Activity*****/
2.  /****临时保存数据*****/
3.      //重写onSaveInstanceState 方法
4.
5.      String tempData = "someting temp";
6.      outState.putString("key",tempData);
7.      /****取回数据*****/
8.
9.      // 在onCreat方法中, Bundle savedInstanceState
10.     if(savedInstanceState!=null){
11.         String tempData = savedInstanceState.getString("key");
12.         Log.d("FirstActivity",tempData);
13.     }

```

● 活动的启动模式

活动模式	解释
standard	默认的模式，系统不会在乎返回栈里有没有这个活动，直接创建一个新的实例
singleTop	系统看看返回栈 顶 有没有这个活动，有就直接使用
singleTask	系统看看返回栈 里 有没有这个活动，有就直接使用
singleInstance	给设置了这个属性的活动一个单独的返回栈，所有的调用都直接使用这个返回栈