



河北师范大学软件学院  
Software College of Hebei Normal University



# Android 基础开发

---

## 第四章 第一节 Activity创建及跳转



Java与移动智能设备开发



# 教学目标

- 掌握在Android中建立Activity的方法
- 掌握在Activity之间传递数据的方式



# 目录



1 Activity概述

2 创建新的Activity

3 Activity与Activity之间的跳转



# Activity概念

- Activity是Android应用中最重要核心组件，每一个应用屏幕就是一个Activity；这意味着，要创建多屏幕的应用，必须创建多个Activity。





# Activity概念

- Android中的Activity的使用基本上分为以下三大类问题：
  - 如何创建多屏幕（ 如何创建多个Activity ）？
  - 屏幕与屏幕之间如何切换（ Activity之间的跳转 ）？
  - 屏幕是何时产生何时消亡的（ Activity的生命周期 ）？



# 目录



- 1 Activity概述
- 2 创建新的Activity
- 3 Activity与Activity之间的跳转



# 创建新的Activity

- 创建新的Activity的基本流程是：
  - 创建新的类继承Activity 类（src/指定包/目录下）。
  - 为该Activity类绑定布局（res/layout/目录下）。
  - 在AndroidManifest.xml文件中注册该Activity。



# 创建新的Activity



- 简单通讯录
  - 创建联系人详细信息显示页面







# 创建新的Activity

- Step1 : 在src目录下建立新的Activity类

```
public class ShowUserActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // TODO Auto-generated method stub  
        super.onCreate(savedInstanceState);  
    }  
}
```

- 新的Activity类必须继承Activity类。
- 必须实现onCreate( )回调方法（创建Activity时自动调用，相当于构造方法）。



# 创建新的Activity

- Step2 : 为新的Activity绑定视图页面
  - 在res/layout/目录下建立新的xml布局文件
  - 为Activity绑定该视图文件

```
//为当前Activity绑定视图页面  
setContentView(R.layout.show_user);
```



# 创建新的Activity

- Step3 : 在AndroidManifest.xml文件中注册Activity

```
<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity android:name=".AddressBookActivity"
        android:label="@string/app_name" >
        <intent-filter>□
    </activity>
    <activity android:name=".ShowUserActivity" />
</application>
```

– 创建一个新的Activity，必须在AndroidManifest.xml文件中注册，否则此Activity不可使用。



# 创建新的Activity

- Step3 : 在AndroidManifest.xml文件中注册Activity
  - 声明的基本语法 :

```
<manifest ... >
  <application ... >
    <activity android:name=".ExampleActivity" />
    ...
  </application ... >
  ...
</manifest >
```

- android:name属性表示当前Activity对应的Java类名。



# 创建新的Activity

- Step3 : 在AndroidManifest.xml文件中注册Activity
  - 声明的基本语法：
    - 在<activity>元素之间，经常需要添加<intent-filter>元素，以确保其它Activity是否可以启动当前Activity（具体内容在第5章中涉及）。
    - <activity>元素的常用属性及其子元素详细说明，参考：

<http://developer.android.com/reference/android/R.styleable.html#AndroidManifestActivity>



# 创建Activity小结

- Activity表示应用中的一个屏幕，一个应用中可能有多个屏幕，即对应着多个Activity。
- 创建新的Activity的基本流程：
  - 创建新的类继承Activity 类（src/指定包/目录下）。
  - 为该Activity类绑定布局（res/layout/目录下）。
  - 在AndroidManifest.xml文件中注册该Activity。



# 目录



- 1 Activity概述
- 2 创建新的Activity
- 3 Activity与Activity之间的跳转**



# Activity跳转简介

- 一个Android应用中包含多个Activity，Activity之间必然存在某种跳转关系。







# Activity跳转的基本原理

- 回顾Web应用中，页面与页面之间的跳转是通过HTTP协议进行的。
  - 请求页面会向目的地页面发送一个HTTP请求（借助请求对象实现）。
    - ✓ 请求对象中内容包括（请求目的地、提交数据等）
  - 目的地页面会反馈给请求页面一个HTTP响应（借助响应对象实现）。
    - ✓ 响应对象中包括（响应消息内容）
- Android中要实现页面跳转，同样需要具备充当请求对象和响应对象的东西。



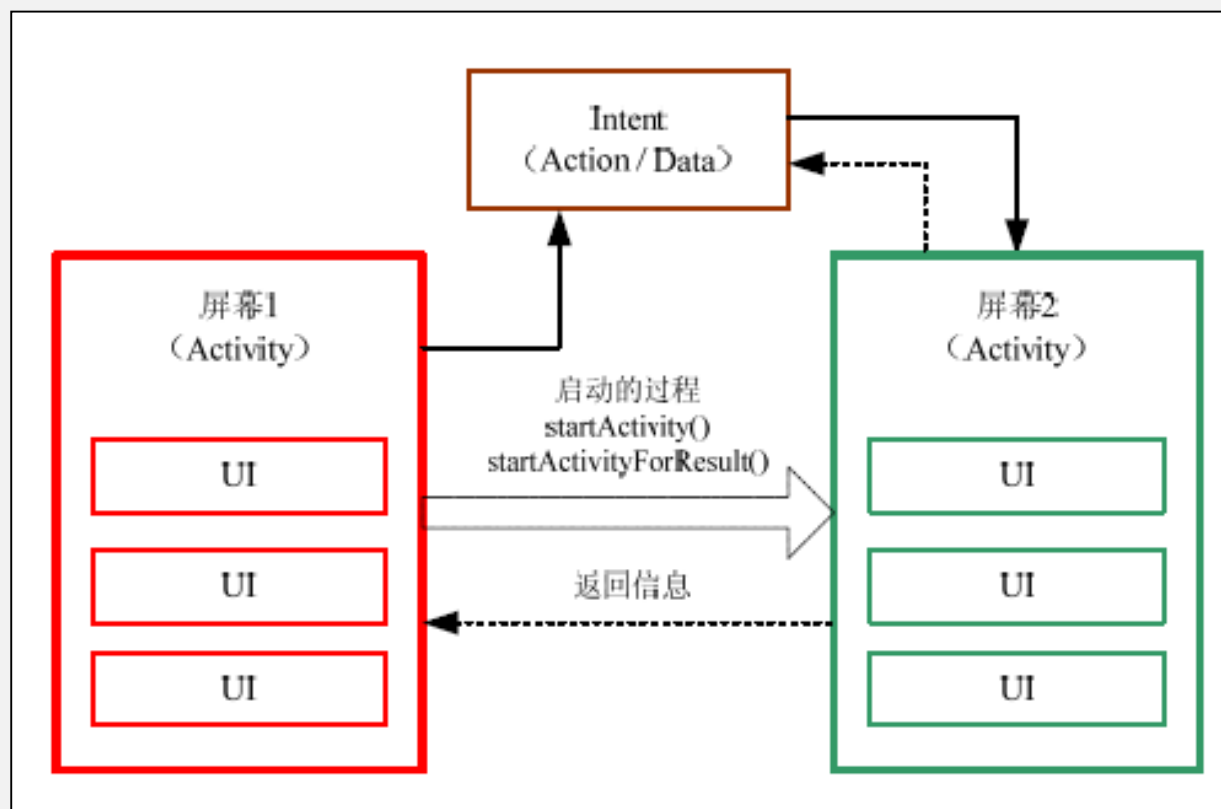
# Activity跳转的基本原理

- 在Android中，Activity与Activity之间的跳转是借助Intent对象来实现的。
  - Intent对象用来在Activity与Activity之间传递请求消息和响应消息。
  - 也就是说，Intent对象充当了HTTP协议中的请求对象和响应对象双重作用。



# Activity跳转的基本原理

- 在Android中，Activity与Activity之间的跳转是借助Intent对象来实现的。





# Intent对象简介

- Intent：意图对象，Activity与Activity之间跳转的中介。
- 如何实现由一个Activity跳转到下一个Activity？
  - 跳转目的地？是否携带参数？
- 目的Activity如何实现向上一个Activity返回数据？



# Activity跳转

- 发送请求的Activity页面
  1. 创建Intent对象：
    - `Intent i = new Intent( );`
  2. 设置请求目的地：
    - `i.setClass( 上下文, 待启动的Activity.class);`
    - `i.setAction( 目的Activity字符串 );`
  3. 携带数据（可选）
  4. 发送请求（启动新的Activity）：
    - `startActivity( Intent对象 );`
    - `startActivityForResult( Intent对象, 请求码 );`



# Activity跳转（携带数据）

- 发送请求的Activity页面
  - 3. 携带数据（可选）：
    - 直接添加基本类型参数：`i.putExtra( key, value );`
    - 传递类的对象（需要序列化对象后才可传递）



# Activity跳转（携带数据）

- 发送请求的Activity页面
  - 3. 携带数据（可选）：
    - 创建复杂数据对象：借助Bundle对象实现
      - » 创建Bundle对象：`Bundle b = new Bundle( );`
      - » 为Bundle对象添加数据：`b.putString( );`、`b.putSerializable( );`、.....
      - » 把Bundle对象添加到Intent对象中：`i.putExtra(Bundle 对象);`

<http://developer.android.com/reference/android/os/Bundle.html>



# Activity跳转（被请求页面处理请求）



- 被请求的Activity页面
  - 获得Intent对象（请求对象）：
    - `Intent request = getIntent( );`
  - 获得请求参数：
    - `request.getIntExtra( );`：返回基本int类型数据
    - `request.getExtras( );`：返回Bundle对象

<http://developer.android.com/reference/android/content/Intent.html>





# Activity跳转（无响应）

- 完成Activity的跳转的步骤：
  - 注册跳转的触发事件；
  - 构造跳转Intent，加入参数；
  - 进行activity的跳转；
  - 在跳转到的activity中接受传入的参数。



# Activity跳转（被请求页面设置响应）



- 被请求的Activity页面
  - 获得Intent对象（响应对象）：
    - `Intent response = new Intent();`
  - 添加响应消息：
    - `response.putIntExtra();`：添加基本int类型数据
    - `response.putExtras();`：添加Bundle对象
  - 实现响应：
    - `this.setResult( int resoponseCode, Intent 响应对象)`



# Activity跳转（请求页获取响应消息）



- 请求的Activity页面

```
// 请求页面接收响应并处理响应消息
protected void onActivityResult(
    int requestCode,    // 请求码
    int resultCode,    // 响应码
    Intent data         // Intent响应对象
) {
    // 接收响应消息
    int i = data.getIntExtra("age", 0);
    Bundle b = data.getExtras();
    // 处理响应消息
}
```

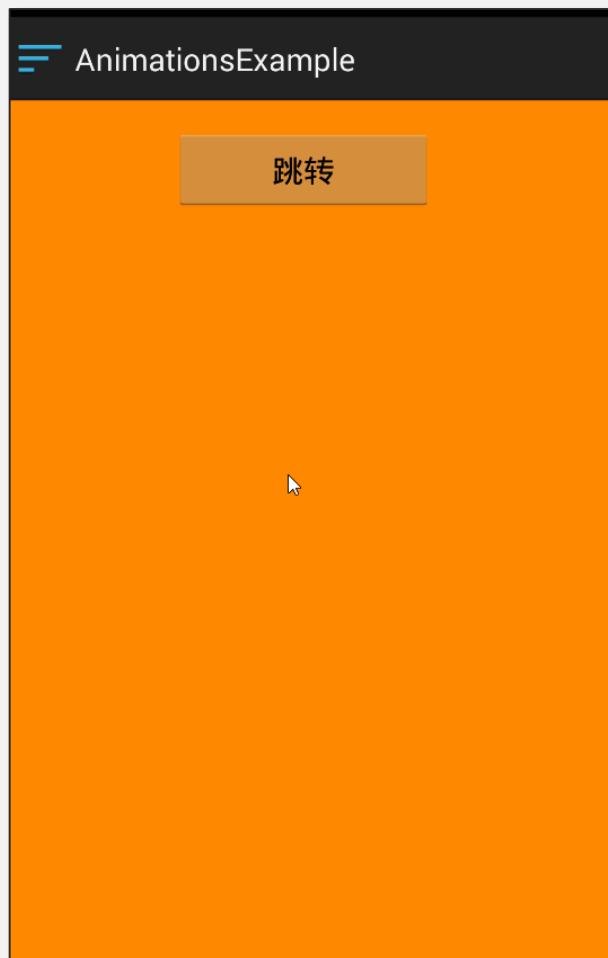


# Activity跳转小结

- Activity跳转：实现屏幕与屏幕之间的切换。
- Activity实现跳转基本流程：
  - 请求页面创建Intent对象
  - 请求页面发送Intent请求（可能添加请求参数）
  - 被请求页面处理请求消息
  - 被请求页面返回响应消息（可选）
    - 请求页面处理响应消息



# 补充练习





# 内容回顾

- 1 Activity概述
- 2 创建新的Activity
- 3 Activity与Activity之间的跳转



Thank you!

