



Android 基础开发

第五章 第一节 Intent的使用





教学目标

- 掌握Intent的基本用法



目录



1 Intent概述

2 使用显示Intent

3 使用隐式Intent



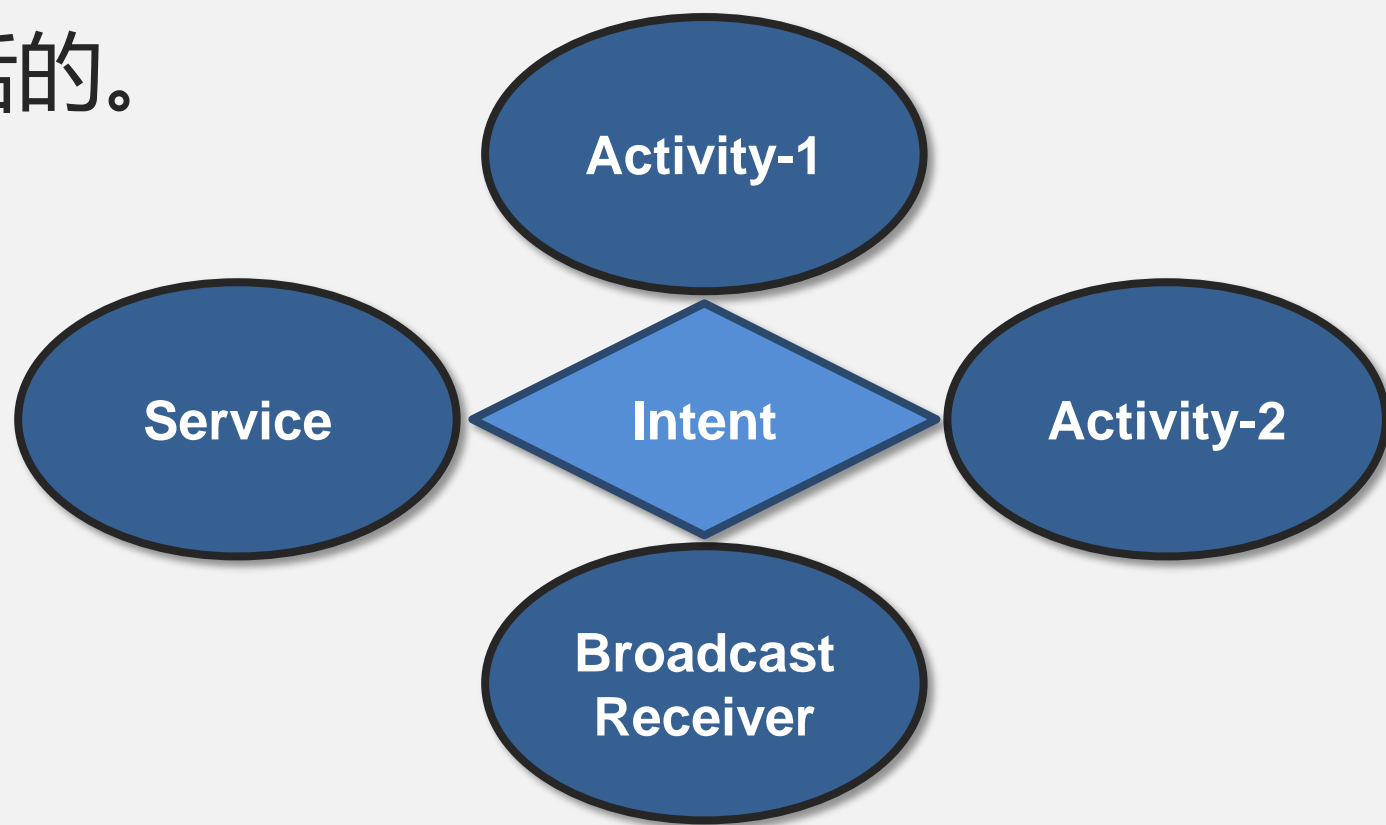
Intent简介

- Activity基本设计理念是：减少组件间的耦合；因此在Android中提供了Intent来实现在应用程序组件与组件之间交互。
- Intent：意图，负责对应用中一次操作的动作、动作涉及数据、附加数据进行描述；Android根据此Intent的描述，负责找到对应的组件，将Intent传递给调用的组件，并完成组件的调用。
- Intent最主要的作用就是，使用**Intent启动三大核心组件**。



Intent简介

- Android中的三大核心组件，**活动(Activity)**、**服务(Service)**和**广播接收器(BroadcastReceiver)**，都是通过Intent来启动或激活的。





Intent简介

- 使用Intent启动Activity

方法名称	描述
Context.startActivity()	启动新的Activity
Activity.startActivityForResult()	启动新的Activity(并获得相应信息)

- 使用Intent启动Service

方法名称	描述
Context.startService()	启动新的Service
Context.bindService()	在调用组件和目标Service间建立连接

- 使用Intent启动BroadcastReceiver

方法名称	描述
Context.sendBroadcast()	普通的发送广播
Context.sendOrderedBroadcast()	有序的发送广播
Context.sendStickyBroadcast()	粘性的发送广播



Intent简介

- Intent由以下各个组成部分：
 - **Component**（组件）：目的组件。
 - **Action**（动作）：用来表现意图的行动。
 - **Category**（类别）：用来表现动作的类别。
 - **Data**（数据）：表示与动作要操纵的数据。
 - **Type**（数据类型）：对于data范例的描写。
 - **Extras**（扩展信息）：扩展信息。
 - **Flags**（标志位）：期望这个意图的运行模式。



Intent简介

- Intent对象在启动组件时，需要明确一个核心问题：
 - 当前Intent对象待启动的组件是什么？
- 根据这个问题的解决思路，一般在Android中有两种形式的Intent使用方法：
 - **显示Intent对象**：直接指明Intent的目的组件
 - **隐式Intent对象**：没有直接指明Intent目的组件，而是通过Intent其它属性来隐式指出Intent待加载的组件类



目录



- 1 Intent概述
- 2 使用显示Intent
- 3 使用隐式Intent



使用显式Intent

- 显式Intent：明确指定了当前Intent应该传递给哪个组件的Intent。
- 显式Intent使用的基本方法：
 - **setClass**(Context packageContext, Class<?> cls)
 - **setClassName**(Context packageContext, String className)
 - **setClassName**(String packageName, String className)



使用显式Intent

- Component(组件)：目的组件，明确指定Intent的目标组件的类名称，Component属性需要接受一个ComponentName对象。

- **setComponent**(ComponentName component)

- ComponentName对象的构造器如下：

- **ComponentName**(String pkg, String cls)
- **ComponentName**(Context pkg, String cls)
- **ComponentName**(Context pkg, Class<?> cls)



使用显式Intent

- 当需要为Intent设置Component属性时，Intent已经提供了一个简化的构造器，这样可以方便指定启动其他组件。

- **Intent**(Context packageContext, Class<?> cls)

- 在组件中可以使用如下方法获取Intent，并且通过Intent获取组件包名及组件类名。

类	方法名称	描述
Activity	getIntent()	获取当前Intent
	getComponentName()	获取ComponentName
Intent	getComponent()	获取ComponentName
ComponentName	getPackageName()	获取组件包名
	getClassName()	获取组件类名



目录



- 1 Intent概述
- 2 使用显式Intent
- 3 使用隐式Intent



隐式Intent

- 隐式Intent：未明确指明待加载的组件类，但是在Intent属性中给出了待加载组件需要满足的特征。那么对应被启动的组件中接受此种属性。
- 组件的特征取决于在Activity配置中<intent-filter.../>元素的配置。<intent-filter.../>就是用于描述Intent的各种属性，<intent-filter.../>元素通常可包含如下子元素。
 - 0~N个 <action.../> 子元素
 - 0~N个 <category.../> 子元素
 - 0~1个 <data.../> 子元素



Intent的Action、Category属性

- Action和Category属性都是普通的字符串，Action代表想要完成的一个“动作”，Category代表此动作的额外信息。
- 被启动的组件是不明确的，取决于Activity的<intent-filter.../>属性。例如：我提供的Action是="com.example.intent.AAA"，那么就启动指定intent-filter的组件。Action和Category属性要结合使用。

```
<action android:name="com.example.intent.AAA"/>  
<category android:name="android.intent.category.DEFAULT"/>
```



Intent的Action、Category属性

- Android内部提供了大量的标准Action、Category常量串的组合。

Action 常量	对应字符串	简单说明
ACTION_MAIN	android.intent.action.MAIN	应用程序入口
ACTION_VIEW	android.intent.action.VIEW	显示指定数据
ACTION_ATTACH_DATA	android.intent.action.ATTACH_DATA	指定某块数据将被附加到其他地方
ACTION_EDIT	android.intent.action.EDIT	编辑指定数据
ACTION_PICK	android.intent.action.PICK	从列表中选择某项并返回所选的数据
ACTION_CHOOSER	android.intent.action.CHOOSER	显示一个 Activity 选择器
ACTION_GET_CONTENT	android.intent.action.GET_CONTENT	让用户选择数据，并返回所选数据
ACTION_DIAL	android.intent.action.DIAL	显示拨号面板
ACTION_CALL	android.intent.action.CALL	直接向指定用户打电话
ACTION_SEND	android.intent.action.SEND	向其他人发送数据
ACTION_SENDTO	android.intent.action.SENDTO	向其他人发送消息
ACTION_ANSWER	android.intent.action.ANSWER	应答电话
ACTION_INSERT	android.intent.action.INSERT	插入数据



Intent的Action、Category属性

Category 常量	对应字符串	简单说明
CATEGORY_DEFAULT	android.intent.category.DEFAULT	默认的 Category
CATEGORY_BROWSABLE	android.intent.category.BROWSABLE	指定该 Activity 能被浏览器安全调用
CATEGORY_TAB	android.intent.category.TAB	指定 Activity 作为 TabActivity 的 Tab 页
CATEGORY_LAUNCHER	android.intent.category.LAUNCHER	Activity 显示顶级程序列表中
CATEGORY_INFO	android.intent.category.INFO	用于提供包信息
CATEGORY_HOME	android.intent.category.HOME	设置该 Activity 随系统启动而运行
CATEGORY_PREFERENCE	android.intent.category.PREFERENCE	该 Activity 是参数面板
CATEGORY_TEST	android.intent.category.TEST	该 Activity 是一个测试
CATEGORY_CAR_DOCK	android.intent.category.CAR_DOCK	指定手机被插入汽车底座（硬件）时运行该 Activity
CATEGORY_DESK_DOCK	android.intent.category.DESK_DOCK	指定手机被插入桌面底座（硬件）时运行该 Activity
CATEGORY_CAR_MODE	android.intent.category.CAR_MODE	设置该 Activity 可在车载环境下使用



Intent的Action、Category属性

- Action属性使用时注意事项：
 1. Intent对象只能设定一个Action属性；表明某一个Intent只能加载一个组件。
 2. 若Intent为隐式Intent，且指明了Action属性，则待加载组件必须在AndroidManifest.xml文件中进行注册，且添加<intent-filter>元素并声明<action>子元素。



Intent的Action、Category属性

3. 在AndroidManifest.xml文件中，一个组件的 `<intent-filter>` 元素中可以包含多个 `<action>` 子元素，表明该组件可以被不同的Intent加载。
4. Action属性值可以使用内置的，也可以使用用户自定义的。
5. 若符合Intent对象的待加载组件有多个，将以列表形式显示出来以供用户选择。



Intent的Action、Category属性

- Category属性使用时注意事项：
 1. Intent对象可以添加多个Category属性；使用addCategory()和removeCategory()来添加或删除属性。
 2. 若Intent为隐式Intent，且指明了Category属性，则待加载组件需要在AndroidManifest.xml文件中进行注册，且添加<intent-filter>元素声明<category>子元素。



Intent的Action、Category属性

3. 在AndroidManifest.xml文件中，一个组件的 `<intent-filter>` 元素中可以包含多个 `<category>` 子元素。
4. `category` 属性值可以使用内置的，也可以使用用户自定义的。
5. 若Intent对象指明了action属性，但未指明category

```
String CATEGORY_DEFAULT  
    = "android.intent.category.DEFAULT"
```



Intent的Data、Type属性

- Data属性通常用于向Action属性提供操作的数据，Data属性接受一个Uri对象，Uri对象通过如下形式的字符串表示：

scheme://host:port/path

- 例如：`content: //com.android.contacts/contacts/1`
 - scheme部分是content
 - host部分是com.android.contacts
 - port部分被省略了
 - path部分是/contacts/1



Intent的Data、Type属性

- Type属性用于指定Data属性的MIME类型，这种MIME类型可以是任何自定义的类型，只要符合abc/xyz格式的字符串即可。
 - 添加data属性使用setData()方法
 - 设置type属性使用setType()方法
- data和type属性一般只需要一个，通过setData方法会把type属性设置为null，相反设置setType方法会把data设置为null，如果想要两个属性同时设置，要使用setDataAndType()方法。



Intent的Data、Type属性

- 例如：Action属性、Data属性的组合。

```
intent.setAction(Intent.ACTION_VIEW);  
Uri data = Uri.parse("http://localhost:8080/index ");  
intent.setData(data);
```

当希望打开满足上述条件的组件时，则需要对组件的<intent-filter>做如下声明。

```
<action android:name="android.intent.action.VIEW" />  
<category android:name="android.intent.category.DEFAULT" />  
<data android:scheme="http"  
        android:host="localhost"  
        android:port="8080"  
        android:path="/index"/>
```




Intent的Data、Type属性

- 例如：Data属性、Type属性同时使用。

```
Intent intent = new Intent(Intent.ACTION_VIEW);  
Uri uri = Uri.parse("file:///mp3/平凡之路.mp3");  
intent.setDataAndType(uri, "audio/mp3");
```

当希望打开满足上述条件的组件时，则需要对组件的<intent-filter>做如下声明。

```
<action android:name="android.intent.action.VIEW" />  
<category android:name="android.intent.category.DEFAULT" />  
<data android:scheme= "file"  
        android:host= "/mp3"  
        android:path= "/平凡之路mp3"  
        android:mimeType= "audio/mp3"/>
```



Intent的Extra属性

- Extra属性通常用于在多个Action之间进行数据交换。Intent的Extra属性值一般是一个Bundle对象，Bundle类类似于Map，存在多组key-value对，这样就可以在多个Activity之间交换数据了。
- 添加数据使用putExtra()进行添加。

```
intent.putExtra("移动", "10086");
```

- 获取数据使用getStringExtra()进行获取。

```
intent.getStringExtra("移动");
```



Intent的Flags属性

- Activity和Task(活动栈)的关系。
 - Task就像一个容器，而Activity就相当于填充这个容器的内容，第一个Activity会处于最下面，最后添加的Activity会在最顶端。从Task中取出Activity是从最顶端取出。
- Flags（标志位）表示Intent的运行模式。



Intent的Flags属性

- Flags (标志位) 表示Intent的运行模式

```
// 如果Activity在Task存在，拿到最顶端，不会启动新的Activity  
intent.addFlags(Intent.FLAG_ACTIVITY_BROUGHT_TO_FRONT);
```

```
// 如果Activity在Task存在，将Activity之上的所有Activity结束掉  
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
// 默认的跳转类型，将Activity放到一个新的Task中  
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

```
// 如果Activity已运行到了Task，再次跳转不会再运行这个Activity  
intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
```



内容回顾

- 1 Intent概述
- 2 使用显式Intent
- 3 使用隐式Intent



Thank you!

