

自助者天助

博客园 首页 新随笔 联系 订阅 管理

随笔 - 324 文章 - 2 评论 - 56

第十章：Intent详解

【正文】

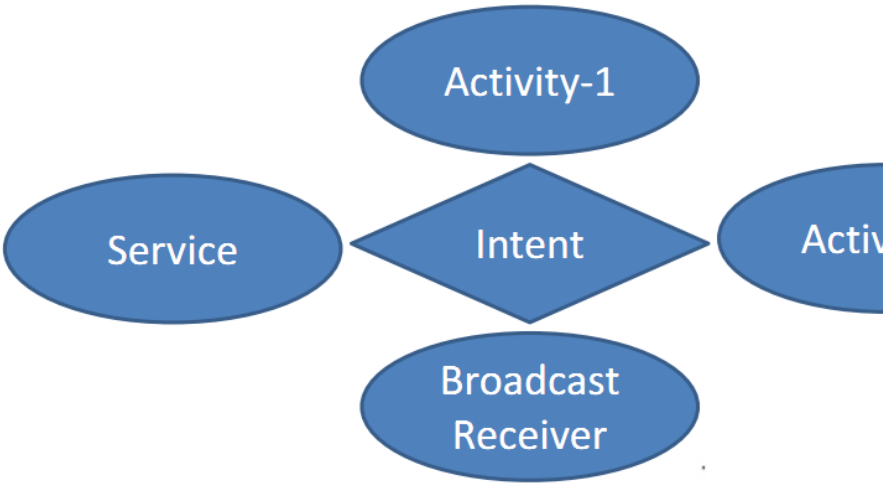
Intent组件虽然不是四大组件，但却是连接四大组件的桥梁，学习好这个知识，也非常的重要。

一、什么是Intent

1、Intent的概念：

- Android中提供了Intent机制来协助应用间的交互与通讯，或者采用更准确的说法是，Intent不仅可用于应用程序之间，也可用于应用程序内部的activity, service和broadcast receiver之间的交互。Intent这个英语单词的本意是“目的、意向、意图”。
- Intent是一种运行时绑定（runtime binding）机制，它能在程序运行的过程中连接两个不同的组件。通过Intent，你的程序可以向Android表达某种请求或者意愿，Android会根据意愿的内容选择适当的组件来响应。

activity、service和broadcast receiver之间是通过Intent进行通信的，而另外一个组件Content Provider本身就是一种通信机制，不需要通过Intent。我们来看下面这个图就知道了：



如果Activity1需要和Activity2进行联系，二者不需要直接联系，而是通过Intent作为桥梁。通俗来讲，Intent类似于中介、媒婆的角色。

2、对于向这三种组件发送intent有不同的机制：

- 使用Context.startActivity() 或 Activity.startActivityForResult(), 传入一个intent来启动一个activity。使用 Activity.setResult(), 传入一个intent来从activity中返回结果。
- 将intent对象传给Context.startService()来启动一个service或者传消息给一个运行的service。将intent对象传给 Context.bindService()来绑定一个service。
- 将intent对象传给 Context.sendBroadcast(), Context.sendOrderedBroadcast(), 或者 Context.sendStickyBroadcast()等广播方法，则它们被传给 broadcast receiver。

二、Intent的相关属性：

- 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

公告

昵称：自助者天助
园龄：9年2个月
粉丝：60
关注：9
[+加关注](#)

< 2016年11月 >						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

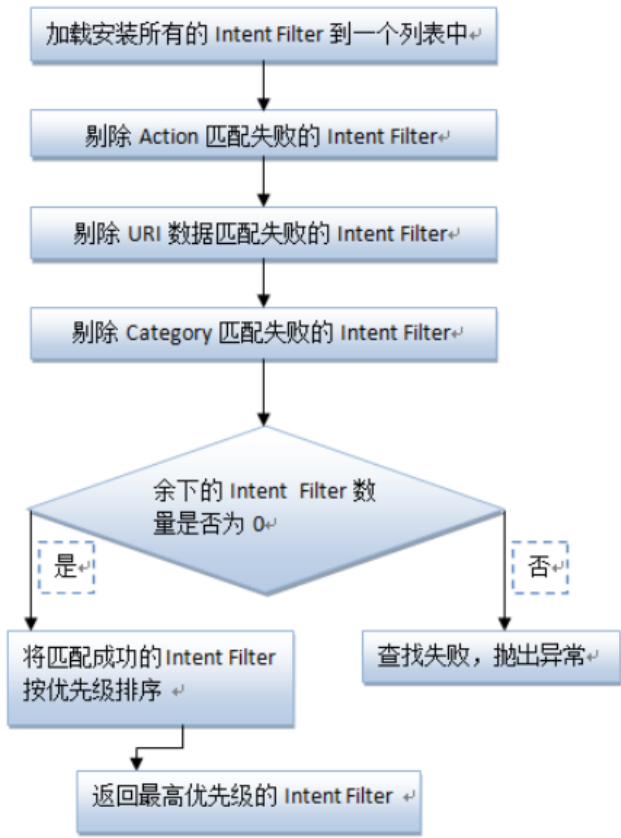
随笔分类(313)

- AJAX(5)
- Android系列(19)
- Best practices(9)
- C#(28)
- C#2008与.NET 3.5 高级程...
- C#与.NET高级程序设计系列...
- CLR(8)
- CSS(1)
- Enterprise Library(1)
- Javascript(17)
- NET Framework(19)
- SQL SERVER(34)
- Test Driven Develop(5)
- UML(7)
- WebService(9)
- XML(2)
- 操作系统(13)
- 从零开始学习jQuery(10)
- 大道至简(18)
- 多线程(10)
- 关键代码
- 加密(1)
- 架构设计(3)
- 经济(1)
- 面向对象(1)
- 其他(11)
- 设计与模式(14)

Intent类型分为显式Intent（直接类型）、隐式Intent（间接类型）。官方建议使用隐式Intent。上述属性中，component属性为直接类型，其他均为间接类型。

相比与显式Intent，隐式Intent则含蓄了许多，它并不明确指出我们想要启动哪一个活动，而是指定一系列更为抽象的action和category等信息，然后交由系统去分析这个Intent，并帮我们找出合适的活动去启动。

Activity 中 Intent Filter 的匹配过程：



1、component(组件)：目的组件

Component属性明确指定Intent的目标组件的类名称。（属于直接Intent）

如果 component这个属性有指定的话，将直接使用它指定的组件。指定了这个属性以后，Intent的其它所有属性都是可选的。

例如，启动第二个Activity时，我们可以这样来写：

```
1 button1.setOnClickListener(new OnClickListener() {
2     @Override
3     public void onClick(View v) {
4         //创建一个意图对象
5         Intent intent = new Intent();
6         //创建组件，通过组件来响应
7         ComponentName component = new ComponentName(MainActivity.this,
8             SecondActivity.class);
9         intent.setComponent(component);
10        startActivity(intent);
11    }
12 });
```

如果写的简单一点，监听事件onClick()方法里可以这样写：

```
1 Intent intent = new Intent();
2 //setClass函数的第一个参数是一个Context对象
3 //Context是一个类，Activity是Context类的子类，也就是说，所有的Activity对象，都可以向上转型为Context对象
4 //setClass函数的第二个参数是一个Class对象，在当前场景下，应该传入需要被启动的
```

数据库性能调优系列(4)

数据库原理与设计(9)

项目管理(5)

小村系列(19)

随笔档案(324)

2015年1月 (4)

2014年12月 (8)

2014年11月 (7)

2014年3月 (1)

2013年12月 (4)

2013年11月 (7)

2013年10月 (4)

2013年9月 (40)

2013年1月 (6)

2012年3月 (1)

2012年2月 (10)

2010年9月 (2)

2010年8月 (10)

2010年7月 (9)

2010年6月 (4)

2010年5月 (3)

2010年4月 (1)

2010年3月 (4)

2009年12月 (4)

2009年11月 (1)

2009年9月 (2)

2009年8月 (4)

2009年7月 (4)

2009年6月 (9)

2009年5月 (5)

2009年4月 (17)

2009年3月 (10)

2009年1月 (1)

2008年12月 (5)

2008年11月 (2)

2008年10月 (2)

2008年9月 (27)

2008年8月 (36)

2008年4月 (15)

2008年3月 (20)

2008年2月 (3)

2008年1月 (12)

2007年12月 (2)

2007年8月 (18)

最新评论

- 1. Re:第十章：Intent详解
很棒
--正能量制造机
- 2. Re:第十章：Intent详解
看了你写得Intent的详解月案例，深受启发。像这样思路明确条理清晰的好文应该值得推广。
佩服!!!
--王光
- 3. Re:第十五章:Android 调...
哥们，看了你的文章写得很不错，期待你以后的文章
--王光
- 4. Re:第十六章:异步处理之...
哥们，看了你的文章写得很不错，期待你以后的文章
--王光
- 5. Re:第十七章:android解...
哥们，看了你的文章写得很不错，期待你以后的文章
--王光

阅读排行榜

- 1. 第十章：Intent详解(170...
- 2. 进程和线程的区别(7465)

Activity类的class对象

```
5         intent.setClass(MainActivity.this, SecondActivity.class);
6         startActivity(intent);
```

再简单一点, 可以这样写: (当然, 也是最常见的写法)

```
1         Intent intent = new Intent(MainActivity.this, SecondActivity.class);
2         startActivity(intent);
```

2、Action (动作): 用来表现意图的行动

当日常生活中, 描述一个意愿或愿望的时候, 总是有一个动词在其中。比如: 我想“做”三个俯卧撑; 我要“写”一封情书, 等等。在Intent中, Action就是描述做、写等动作的, 当你指明了一个Action, 执行者就会依照这个动作的指示, 接受相关输入, 表现对应行为, 产生符合的输出。在Intent类中, 定义了一批量的动作, 比如ACTION_VIEW, ACTION_PICK等, 基本涵盖了常用动作。加的动作越多, 越精确。

Action 是一个用户定义的字符串, 用于描述一个 Android 应用程序组件, 一个 Intent Filter 可以包含多个 Action。在 AndroidManifest.xml 的Activity 定义时, 可以在其 `<intent-filter>` 节点指定一个 Action列表用于标识 Activity 所能接受的“动作”。

3、category (类别): 用来表现动作的类别

Category属性也是作为<intent-filter>子元素来声明的。例如:

```
<intent-filter>
    <action android:name="com.vince.intent.MY_ACTION"></action>
    <category android:name="com.vince.intent.MY_CATEGORY"></category>
    <category android:name="android.intent.category.DEFAULT"></category>
</intent-filter>
```

Action 和category通常是放在一起用的, 所以这里一起介绍一下。我们来先来举一个例子:

新建一个工程文件smyh006_Intent01, 在默认文件的基础之上, 新建文件SecondActivity.java和activity_second.xml。

紧接着, 我们要到清单文件中进行注册, 打开AndroidManifest.xml, 添加SecondActivity的action和category的过滤器:

```
1         <activity
2             android:name=".SecondActivity">
3             <intent-filter>
4                 <action android:name="com.example.smyh006intent01.MY_ACTION"/>
5                 <category android:name="android.intent.category.DEFAULT" />
6             </intent-filter>
7         </activity>
```

上方代码, 表示SecondActivity可以匹配第4行的MY_ACTION这个动作, 此时, 如果在其他的Activity通过这个action的条件来查找, 那SecondActivity就具备了这个条件。类似于相亲时, 我要求对方有哪些条件, 然后对方这个SecondActivity恰巧满足了这个条件 (够通俗了吧)。

注: 如果没有指定的category, 则必须使用默认的DEFAULT (即上方第5行代码)。

也就是说: 只有<action>和<category>中的内容同时能够匹配上Intent中指定的action和category时, 这个活动才能响应Intent。如果使用的是DEFAULT这种默认的category, 在稍后调用startActivity()方法的时候会自动将这个category添加到Intent中。

现在来修改MainActivity.java中按钮的点击事件, 代码如下:

```
1         button1.setOnClickListener(new OnClickListener() {
2             @Override
3             public void onClick(View v) {
4                 //启动另一个Activity, (通过action属性进行查找)
```

3. XmlElement和XmlNode...

4. 使用Project进行项目管理...

5. SQL 连接 JOIN 例解。(...

评论排行榜

1. 第十章: Intent详解(8)
2. 进程和线程的区别(5)
3. SQL 连接 JOIN 例解。(...
4. SQL Server TransActio...
5. 类与类之间的关系及在代...

推荐排行榜

1. 第十章: Intent详解(20)
2. 进程和线程的区别(3)
3. 第六章: 四大组件之Activ...
4. 第八章: 四大组件之Cont...
5. 第七章: 四大组件之Servi...

```
5         Intent intent = new Intent();
6         //设置动作（实际action属性就是一个字符串标记而已）
7         intent.setAction("com.example.smyh006intent01.MY_ACTION"); //方法:
Intent android.content.Intent.setAction(String action)
8         startActivity(intent);
9     }
10    });
```

上方代码中，也可以换成下面这种简洁的方式：

```
1         button1.setOnClickListener(new OnClickListener() {
2             @Override
3             public void onClick(View v) {
4                 //启动另一个Activity，（通过action属性进行查找）
5                 Intent intent = new Intent("com.example.smyh006intent01.MY_ACTION");//方
法: android.content.Intent.Intent(String action)
6                 startActivity(intent);
7             }
8         });
```

上方第5行代码：在这个Intent中，我并没有指定具体哪一个Activity，我只是指定了一个action的常量。所以说，隐式Intent的作用就表现的淋漓尽致了。此时，点击MainActivity中的按钮，就会跳到SecondActivity中去。

上述情况只有SecondActivity匹配成功。如果有多个组件匹配成功，就会以对话框列表的方式让用户进行选择。我们来详细介绍一下：

我们新建文件ThirdActivity.java和activity_third.xml，然后在清单文件AndroidManifest.xml中添加ThirdActivity的action和category的过滤器：

```
1         <activity
2             android:name=".ThirdActivity">
3             <intent-filter>
4                 <action android:name="com.example.smyh006intent01.MY_ACTION"/>
5                 <category android:name="android.intent.category.DEFAULT" />
6             </intent-filter>
7         </activity>
```

此时，运行程序，当点击MainActivity中的按钮时，弹出如下界面：



相信大家看到了这个界面，应该就一目了然了。于是我们可以做出如下总结：

在自定义动作时，使用**activity**组件时，必须添加一个默认类别

具体的实现为：

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

如果有多个组件被匹配成功，就会以对话框列表的方式让用户进行选择。

每个**Intent**中只能指定一个**action**，但却能指定多个**category**；类别越多，动作越具体，意图越明确（类似于相亲时，给对方提了很多要求）。

目前我们的**Intent**中只有一个默认的**category**，现在可以通过**intent.addCategory()**方法来实现。修改**MainActivity**中按钮的点击事件，代码如下：

```
1 button1.setOnClickListener(new OnClickListener() {
2     @Override
3     public void onClick(View v) {
4         //启动另一个Activity，（通过action属性进行查找）
```

```

5         Intent intent = new Intent();
6         //设置动作 (实际action属性就是一个字符串标记而已)
7         intent.setAction("com.example.smyh006intent01.MY_ACTION"); //方法:
Intent android.content.Intent.setAction(String action)
8         intent.addCategory("com.example.smyh006intent01.MY_CATEGORY");
9         startActivity(intent);
10    }
11    });

```

既然在Intent中增加了一个category, 那么我们要在清单文件中去声明这个category, 不然程序将无法运行。代码如下:

```

1         android:name=".SecondActivity">
2         <intent-filter>
3             <action android:name="com.example.smyh006intent01.MY_ACTION"/>
4             <category android:name="android.intent.category.DEFAULT" />
5             <category android:name="com.example.smyh006intent01.MY_CATEGORY" />
6         </intent-filter>
7     </activity>

```

此时, 点击MainActivity中的按钮, 就会跳到SecondActivity中去。

总结如下:

自定义类别: 在Intent添加类别可以添加多个类别, 那就要求被匹配的组件必须同时满足这多个类别, 才能匹配成功。操作Activity的时候, 如果没有类别, 须加上默认类别

4、data (数据): 表示与动作要操纵的数据

- Data属性是Android要访问的数据, 和action和Category声明方式相同, 也是在<intent-filter>中。
- 多个组件匹配成功显示优先级高的; 相同显示列表。

Data是用一个uri对象来表示的, uri代表数据的地址, 属于一种标识符。通常情况下, 我们使用action+data属性的组合来描述一个意图: 做什么

使用隐式Intent, 我们不仅可以启动自己程序内的活动, 还可以启动其他程序的活动, 这使得Android多个应用程序之间的功能共享成为了可能。比如应用程序中需要展示一个网页, 没有必要自己去实现一个浏览器 (事实上也不太可能), 而是只需要条用系统的浏览器来打开这个网页就行了。

【实例】打开指定网页:

MainActivity.java中, 监听器部分的核心代码如下:

```

1         button1.setOnClickListener(new OnClickListener() {
2             @Override
3             public void onClick(View v) {
4                 Intent intent = new Intent();
5                 intent.setAction(Intent.ACTION_VIEW);
6                 Uri data = Uri.parse("http://www.baidu.com");
7                 intent.setData(data);
8                 startActivity(intent);
9             }
10    });

```

当然, 上方代码也可以简写成:

```

1         button1.setOnClickListener(new OnClickListener() {
2             @Override

```

```
3         public void onClick(View v) {
4             Intent intent = new Intent(Intent.ACTION_VIEW);
5             intent.setData(Uri.parse("http://www.baidu.com"));
6             startActivity(intent);
7         }
8     });
```

第4行代码：指定了Intent的action是 **Intent.ACTION_VIEW**，表示查看的意思，这是一个Android系统内置的动作；

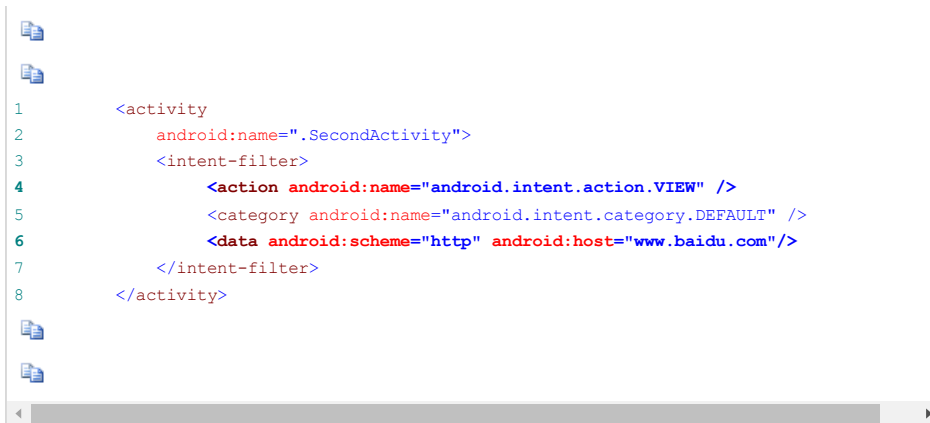
第5行代码：通过Uri.parse()方法，将一个网址字符串解析成一个Uri对象，再调用intent的setData()方法将这个Uri对象传递进去。

当点击按钮时，将跳到如下界面：



此时，调用的是系统默认的浏览器，也就是说，只调用了这一个组件。现在如果有多个组件得到了匹配，应该是什么情况呢？

我们修改修改清单文件中对SecondActivity的声明：



现在，SecondActivity也匹配成功了，我们运行程序，点击MainActivity的按钮时，弹出如下界面供我们选择：



我们可以总结如下：

- 当Intent匹配成功的组件有多个时，显示优先级高的组件，如果优先级相同，显示列表让用户自己选择
- 优先级从-1000至1000，并且其中一个必须为负的才有效

注：系统默认的浏览器并没有做出优先级声明，其优先级默认为正数。

优先级的配置如下：

在清单文件中修改对SecondActivity的声明，即增加一行代码，通过来android:priority设置优先级，如下：

```
1      <activity
2          android:name=".SecondActivity">
3          <intent-filter android:priority="-1">
4              <action android:name="android.intent.action.VIEW" />
5              <category android:name="android.intent.category.DEFAULT" />
6              <data android:scheme="http" android:host="www.baidu.com"/>
7          </intent-filter>
8      </activity>
```

注：

Data属性的声明中要指定访问数据的Uri和MIME类型。可以在<data>元素中通过一些属性来设置：

android:scheme、android:path、android:port、android:mimeType、android:host等，通过这些属性来对应一个典型的Uri格式scheme://host:port/path。例如：<http://www.google.com>。

5、type（数据类型）：对于data范例的描写

如果Intent对象中既包含Uri又包含Type，那么，在<intent-filter>中也必须二者都包含才能通过测试。

Type属性用于明确指定Data属性的数据类型或MIME类型，但是通常来说，当Intent不指定Data属性时，Type属性才会起作用，否则Android系统将会根据Data属性值来分析数据的类型，所以无需指定Type属性。

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Intent.setDataAndType()方法。

【任务】：[data+type属性的使用](#) 【实例】：[播放指定路径的mp3文件](#)。

具体如下：

新建工程文件smyh006_Intent02，MainActivity.java中按钮监听事件部分的代码如下：

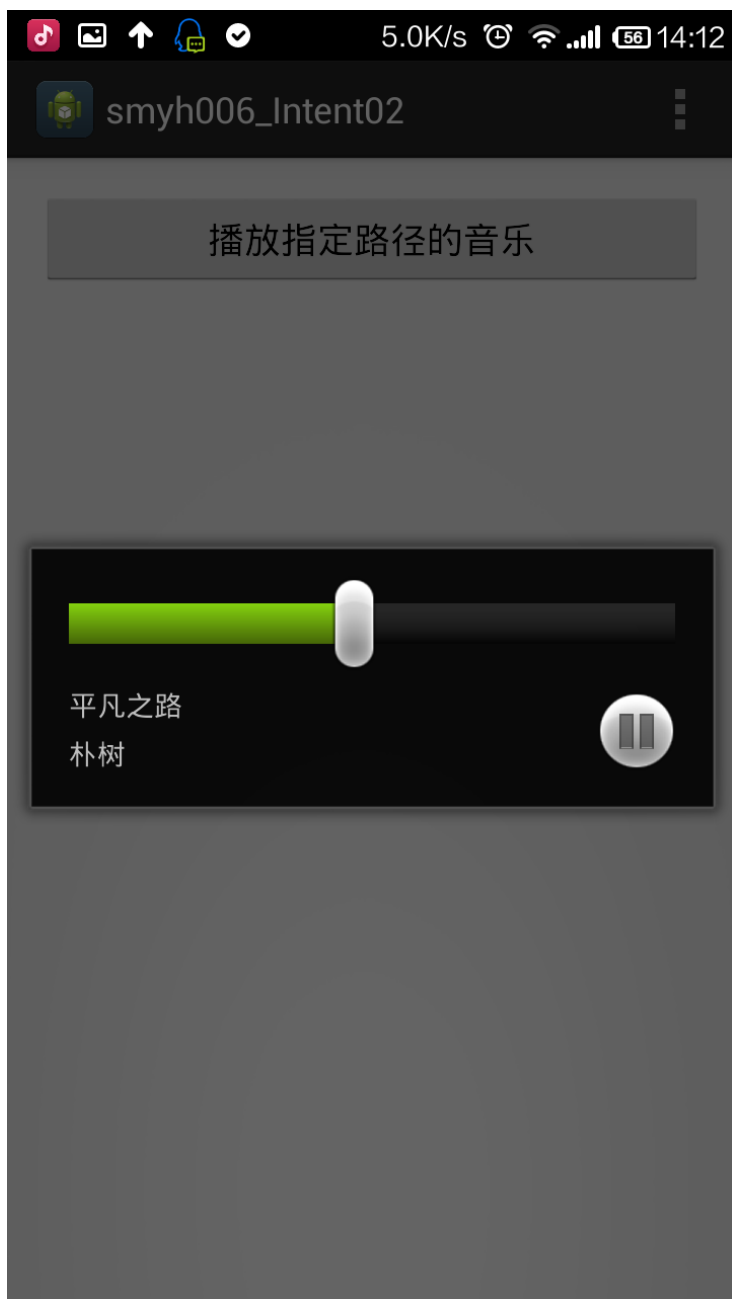
```
1      button.setOnClickListener(new OnClickListener() {
2          @Override
3          public void onClick(View v) {
4              Intent intent = new Intent();
5              intent.setAction(Intent.ACTION_VIEW);
6              Uri data = Uri.parse("file:///storage/sdcard0/平凡之路.mp3");
7              //设置data+type属性
8              intent.setDataAndType(data, "audio/mp3"); //方法: Intent
9              android.content.Intent.setDataAndType(Uri data, String type)
10              startActivity(intent);
11          }
12      });
```

代码解释：

第6行：“file:”表示查找文件，后面再加上我的小米手机存储卡的路径：[/storage/sdcard0](#)，再加上具体歌曲的路径。

第8行：设置data+type属性

运行后，当点击按钮时，效果如下：



上方界面中，使用的是小米系统默认的音乐播放器。

6、extras（扩展信息）：扩展信息

是其它所有附加信息的集合。使用**extras**可以为组件提供扩展信息，比如，如果要执行“发送电子邮件”这个

动作，可以将电子邮件的标题、正文等保存在**extras**里，传给电子邮件发送组件。

7、Flags（标志位）：期望这个意图的运行模式

一个程序启动后系统会为这个程序分配一个**task**供其使用，另外同一个**task**里面可以拥有不同应用程序的**activity**。那么，同一个程序能不能拥有多个**task**？这就涉及到加载**activity**的启动模式，这个需要单独讲一下。

注：android中一组逻辑上在一起的**activity**被叫做**task**，自己认为可以理解成一个**activity**堆栈。

三、Activity的启动模式：（面试注意）

Activity有四种启动模式：**standard**、**singleTop**、**singleTask**、**singleInstance**。可以在**AndroidManifest.xml**中**activity**标签的属性**android:launchMode**中设置该**activity**的加载模式。

- **standard**模式：默认的模式，以这种模式加载时，每当启动一个新的活动，必定会构造一个新的**Activity**实例放到返回栈（目标**task**）的栈顶，不管这个**Activity**是否已经存在于返回栈中；

- **singleTop**模式: 如果一个以**singleTop**模式启动的**activity**的实例已经存在于返回栈的**栈顶**, 那么再启动这个**Activity**时, 不会创建新的实例, 而是重用位于栈顶的那个实例, 并且会调用该实例的 **onNewIntent()**方法将**Intent**对象传递到这个实例中;

注: 如果以**singleTop**模式启动的**activity**的一个实例已经存在于返回栈中, 但是不在栈顶, 那么它的行为和**standard**模式相同, 也会创建多个实例;

- **singleTask**模式: 这种模式下, 每次启动一个**activity**时, 系统首先会在返回栈中检查是否存在该活动的实例, 如果存在, 则直接使用该实例, 并把这个活动之上的所有活动统统清除; 如果没有发现就会创建一个新的活动实例;
- **singleInstance**模式: 总是在新的任务中开启, 并且这个新的任务中有且只有这一个实例, 也就是说被该实例启动的其他**activity**会自动运行于另一个任务中。当再次启动该**activity**的实例时, 会重新调用已存在的任务和实例。并且会调用这个实例的**onNewIntent()**方法, 将**Intent**实例传递到该实例中。和**singleTask**相同, 同一时刻在系统中只会存在一个这样的**Activity**实例。**(singleInstance即单实例)**

注: 前面三种模式中, 每个应用程序都有自己的返回栈, 同一个活动在不同的返回栈中入栈时, 必然是创建了新的实例。而使用**singleInstance**模式可以解决这个问题, 在这种模式下会有一个单独的返回栈来管理这个活动, 不管是哪一个应用程序来访问这个活动, 都公用同一个返回栈, 也就解决了**共享活动实例**的问题。(此时可以实现任务之间的切换, 而不是单独某个栈中的实例切换)

其实我们不在清单文件中设置, 只在代码中通过**flag**来设置也是可以的, 如下:

```
1      Intent intent = new Intent(MainActivity.this,SecondActivity.class);
2      //相当于singleTask
3      intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
4      startActivity(intent);
```

```
1      Intent intent = new Intent(MainActivity.this,SecondActivity.class);
2      //相当于singleTop
3      intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
4      startActivity(intent);
```

三、Intent的常见应用:

1、打开指定网页: (直接复制的上面的代码)

MainActivity.java中, 监听器部分的核心代码如下:

```
1      button1.setOnClickListener(new OnClickListener() {
2          @Override
3          public void onClick(View v) {
4              Intent intent = new Intent();
5              intent.setAction(Intent.ACTION_VIEW); //方法:
6              android.content.Intent(Intent(String action)
7              Uri data = Uri.parse("http://www.baidu.com");
8              intent.setData(data);
9              startActivity(intent);
10         });
```

当然, 上方代码也可以简写成:

```
1      button1.setOnClickListener(new OnClickListener() {
2          @Override
3          public void onClick(View v) {
4              Intent intent = new Intent(Intent.ACTION_VIEW);
5              intent.setData(Uri.parse("http://www.baidu.com"));
6              startActivity(intent);
```

```
7         }  
8     });
```



第4行代码：指定了Intent的action是 **Intent.ACTION_VIEW**，表示查看的意思，这是一个Android系统内置的动作；

第5行代码：通过Uri.parse()方法，将一个网址字符串解析成一个Uri对象，再调用intent的setData()方法将这个Uri对象传递进去。

或者可以写成：



```
1         button1.setOnClickListener(new OnClickListener() {  
2             @Override  
3             public void onClick(View v) {  
4                 Uri uri = Uri.parse("http://www.baidu.com");  
5                 Intent intent = new Intent(Intent.ACTION_VIEW,uri); //方法：  
6                 android.content.Intent(Intent.ACTION_VIEW, uri)  
7                 startActivity(intent);  
8             }  
9         });
```



2、打电话：

【方式一】打开拨打电话的界面：

```
1         Intent intent = new Intent(Intent.ACTION_DIAL);  
2         intent.setData(Uri.parse("tel:10086"));  
3         startActivity(intent);
```

运行程序后，点击按钮，显示如下界面：



【方式二】直接拨打电话：

```
1      Intent intent = new Intent(Intent.ACTION_CALL);
2      intent.setData(Uri.parse("tel:10086"));
3      startActivity(intent);
```

要使用这个功能必须在配置文件中加入权限：（加一行代码）

```
1      <uses-sdk
2          android:minSdkVersion="8"
3          android:targetSdkVersion="16" />
4      <uses-permission android:name="android.permission.CALL_PHONE"/>
```

3、发送短信：

【方式一】打开发送短信的界面：action+type

```

1      Intent intent = new Intent(Intent.ACTION_VIEW);
2      intent.setType("vnd.android-dir/mms-sms");
3      intent.putExtra("sms_body", "具体短信内容"); //"sms_body"为固定内容
4      startActivity(intent);

```

【方式二】打开发短信的界面(同时指定电话号码): **action+data**

```

1      Intent intent = new Intent(Intent.ACTION_SENDTO);
2      intent.setData(Uri.parse("smsto:18780260012"));
3      intent.putExtra("sms_body", "具体短信内容"); //"sms_body"为固定内容
4      startActivity(intent);

```

4、播放指定路径音乐: **action+data+type**

```

1      Intent intent = new Intent(Intent.ACTION_VIEW);
2      Uri uri = Uri.parse("file:///storage/sdcard0/平凡之路.mp3"); ///路径也可以写
成: "/storage/sdcard0/平凡之路.mp3"
3      intent.setDataAndType(uri, "audio/mp3"); //方法: Intent
android.content.Intent.setDataAndType(Uri data, String type)
4      startActivity(intent);

```

5、卸载程序: **action+data** (例如点击按钮, 卸载某个应用程序, 根据包名来识别)

注: 无论是安装还是卸载, 应用程序是根据**包名package**来识别的。

```

1      Intent intent = new Intent(Intent.ACTION_DELETE);
2      Uri data = Uri.parse("package:com.example.smyh006intent01");
3      intent.setData(data);
4      startActivity(intent);

```

6、安装程序: **action+data+type**

```

1      Intent intent = new Intent(Intent.ACTION_VIEW);
2      Uri data = Uri.fromFile(new
File("/storage/sdcard0/AndroidTest/smyh006_Intent01.apk"));    //路径不能写
成: "file:///storage/sdcard0/..."
3      intent.setDataAndType(data, "application/vnd.android.package-archive"); //Type
的字符串为固定内容
4      startActivity(intent);

```

注: 第2行的路径不能写成: "file:///storage/sdcard0/...", 不然报错如下:



疑问: 通过下面的这种方式安装程序, 运行时为什么会出错呢?

```

1      //通过指定的action来安装程序
2      public void installClickTwo(View view){
3          Intent intent = new Intent(Intent.ACTION_PACKAGE_ADDED);
4          Uri data = Uri.fromFile(new
File("/storage/sdcard0/AndroidTest/smyh006_Intent01.apk"));    //路径不能写
成: "file:///storage/sdcard0/..."
5          intent.setData(data);
6          startActivity(intent);
7      }

```



综上所述, 完整版代码如下:



```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:paddingBottom="@dimen/activity_vertical_margin"
6     android:paddingLeft="@dimen/activity_horizontal_margin"
7     android:paddingRight="@dimen/activity_horizontal_margin"
8     android:paddingTop="@dimen/activity_vertical_margin"
9     android:orientation="vertical"
10    tools:context=".MainActivity" >
11    <Button
12        android:id="@+id/button1_browsePage"
13        android:layout_width="match_parent"
14        android:layout_height="wrap_content"
15        android:onClick="browsePageClick"
16        android:text="打开指定网页"/>
17    <Button
18        android:id="@+id/button2_openDialPage"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:onClick="openDialPageClick"
22        android:text="打开拨号面板"/>
23    <Button
24        android:id="@+id/button3_dialPhone"
25        android:layout_width="match_parent"
26        android:layout_height="wrap_content"
27        android:onClick="dialPhoneClick"
28        android:text="直接拨打指定号码"/>
29    <Button
30        android:id="@+id/button4_openMsgPage"
31        android:layout_width="match_parent"
32        android:layout_height="wrap_content"
33        android:onClick="openMsgPageClick"
34        android:text="打开发短信的界面"/>
35
36
37    <Button
38        android:id="@+id/button5_sendMsg"
39        android:layout_width="match_parent"
40        android:layout_height="wrap_content"
41        android:onClick="sendMsgClick"
42        android:text="给指定的人发短信"/>
43
44    <Button
45        android:id="@+id/button6_playMusic"
46        android:layout_width="match_parent"
47        android:layout_height="wrap_content"
48        android:onClick="playMusicClick"
49        android:text="播放指定路径音乐"/>
50
51    <Button
52        android:id="@+id/button7_uninstall"
53        android:layout_width="match_parent"
54        android:layout_height="wrap_content"
55        android:onClick="uninstallClick"
56        android:text="卸载程序"/>
57    <Button
58        android:id="@+id/button8_install"
59        android:layout_width="match_parent"
60        android:layout_height="wrap_content"
61        android:onClick="installClick"
62        android:text="安装程序"/>
```

```
63
64
65 </LinearLayout>
```



MainActivity.java代码如下:

```
1 package com.example.m06intent01;
2 import java.io.File;
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.net.Uri;
6 import android.os.Bundle;
7 import android.view.Menu;
8 import android.view.View;
9 public class MainActivity extends Activity {
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_main);
14     }
15     //打开指定网页
16     public void browsePageClick(View view){
17         Intent intent = new Intent(Intent.ACTION_VIEW);
18         intent.setData(Uri.parse("http://www.baidu.com/"));
19         startActivity(intent);
20     }
21 }
22
23 //打开拨号面板
24 public void openDialPageClick(View view){
25     Intent intent = new Intent(Intent.ACTION_DIAL);
26     intent.setData(Uri.parse("tel:10086"));
27     startActivity(intent);
28 }
29
30 //直接拨打指定号码
31 public void dialPhoneClick(View view){
32     Intent intent = new Intent(Intent.ACTION_CALL);
33     intent.setData(Uri.parse("tel:10086"));
34     startActivity(intent);
35 }
36
37 //打开发短信的界面:action+type
38 public void openMsgPageClick(View view){
39     Intent intent = new Intent(Intent.ACTION_VIEW);
40     intent.setType("vnd.android-dir/mms-sms");
41     intent.putExtra("sms_body", "具体短信内容"); // "sms_body"为固定内容
42     startActivity(intent);
43 }
44
45 //打开发短信的界面(指定电话号码):action+data
46 public void sendMsgClick(View view){
47     Intent intent = new Intent(Intent.ACTION_SENDTO);
48     intent.setData(Uri.parse("smsto:18780260012"));
49     intent.putExtra("sms_body", "具体短信内容"); // "sms_body"为固定内容
50     startActivity(intent);
51 }
52
53 //播放指定路径音乐
54 public void playMusicClick(View view){
55     Intent intent = new Intent(Intent.ACTION_VIEW);
56     Uri uri = Uri.parse("file:///storage/sdcard0/平凡之路.mp3"); //路径也可以写成: "/storage/sdcard0/平凡之路.mp3"
57     intent.setDataAndType(uri, "audio/mp3"); //方法: Intent
58     android.content.Intent.setDataAndType(Uri data, String type)
59     startActivity(intent);
60 }
61 //卸载某个应用程序, 根据包名来识别
62 public void uninstallClick(View view){
63     Intent intent = new Intent(Intent.ACTION_DELETE);
```



```
64     Uri data = Uri.parse("package:com.example.smyh006intent01");
65     intent.setData(data);
66     startActivity(intent);
67 }
68
69 //安装某个应用程序, 根据apk的文件名来识别
70 public void installClick(View view){
71     Intent intent = new Intent(Intent.ACTION_VIEW);
72     Uri data = Uri.fromFile(new
File("/storage/sdcard0/AndroidTest/smyh006_Intent01.apk"));    //路径不能写
成: "file:///storage/sdcard0/..."
73     intent.setDataAndType(data, "application/vnd.android.package-archive"); //Type
的字符串为固定内容
74     startActivity(intent);
75 }
76
77
78 @Override
79 public boolean onCreateOptionsMenu(Menu menu) {
80     // Inflate the menu; this adds items to the action bar if it is present.
81     getMenuInflater().inflate(R.menu.main, menu);
82     return true;
83 }
84
85 }
```



运行后, 主界面如下:



分类: [Android系列](#)

好文要顶

关注我

收藏该文

自助者天助
关注 - 9
粉丝 - 60
[+ 加关注](#)

20

0

« 上一篇: [第九章：四大组件之Broadcast Receiver](#)
» 下一篇: [第十一章：Android数据存储（上）](#)

posted @ 2014-12-05 14:37 自助者天助 阅读(17097) 评论(8) 编辑 收藏

评论列表

#1楼 2015-02-14 17:25 电路格格

写得如此好却没有人点个赞，我来点个赞。

支持(6) 反对(0)

#2楼	2015-10-10 11:54 李_鹏 good	支持(0) 反对(0)
#3楼	2015-12-28 15:09 瘦嘟嘟 Intern还有优先级？	支持(0) 反对(0)
#4楼	2016-06-13 21:26 菜牛一枚 醍醐灌顶！	支持(0) 反对(0)
#5楼	2016-08-09 17:27 吾乃轩辕 博主继续写啊可以出书了，代码分析很细致，很适合我这样的初学者	支持(0) 反对(0)
#6楼	2016-10-11 16:34 liwensb 楼主，写得好，仔细看完了，大受裨益； 所以一定要注册帐号说一声：谢谢！	支持(0) 反对(0)
#7楼	2016-10-11 23:27 王光 看了你写得Intent的详解月案例，深受启发。像这样思路明确条理清晰的好文应该值得推广。 佩服！！！！	支持(0) 反对(0)
#8楼	2016-10-14 16:44 正能量制造机 很棒	支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

最新IT新闻：

- 支付宝在南京等城市开通在线缴纳社保的服务
 - 碎片化并未减轻 发布两个月的安卓7.0占有率仅0.3%
 - 堪比美剧的美国大选，这次Google帮你追定了
 - Facebook开发人才招聘平台，抢占LinkedIn市场
 - 为了不影响看风景，英国人决定把发电厂建在海底
- » 更多新闻...

最新知识库文章：

- 循序渐进地代码重构
 - 技术的正宗与野路子
 - 陈皓：什么是工程师文化？
 - 没那么难，谈CSS的设计模式
 - 程序猿媳妇儿注意事项
- » 更多知识库文章...