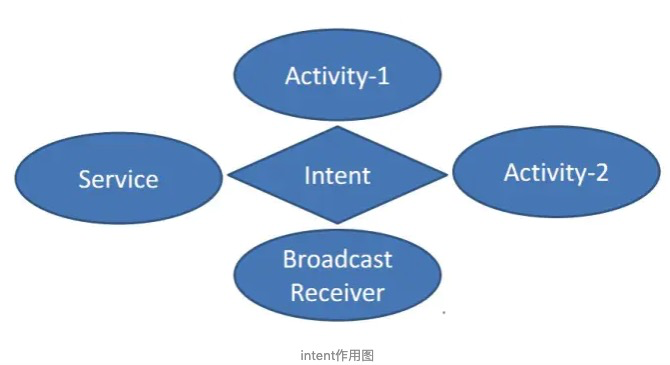
Android Intent用法总结

Android中提供了Intent机制来协助应用间的交互与通讯，Intent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则根据此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不仅可用于应用程序之间，也可用于应用程序内部的 Activity / Service之间的交互。因此，Intent在这里起着一个媒体中介的作用，专门提供组件互相调用的相关信息，实现调用者与被调用者之间的解耦。



##### 1. Intent作用

Intent是一个将要执行的动作的抽象的描述，一般来说是作为参数来使用，由 Intent来协助完成 Android各个组件之间的通讯。比如说调用startActivity()来启动一个Activity，或者由broadcaseIntent()来传递给所有感兴趣的BroadcaseReceiver，再或者由startService() / bindservice()来启动一个后台的 service。所以可以看出来，Intent 主要是用来启动其他的 activity 或者 service，所以可以将 intent 理解成 activity 之间的粘合剂。

Intent作用的表现形式为：

* **启动Activity**  
  通过Context.startActvity() / Activity.startActivityForResult()启动一个Activity；
* **启动Service**  
  通过Context.startService()启动一个服务，或者通过Context.bindService()和后台服务交互；
* **发送Broadcast**  
  通过广播方法Context.sendBroadcasts() / Context.sendOrderedBroadcast() / Context.sendStickyBroadcast()发给Broadcast Receivers

##### 2. Intent种类

* **显式Intent**  
  显式，即直接指定需要打开的activity对应的类。

1）构造方法传入Component，最常用的方式：

Intent intent = new Intent(this, SecondActivity.class); startActivity(intent);

2）setComponent方法

Intent intent = new Intent();

intent.setClass(this, SecondActivity.class); //或者intent.setClassName(this, "com.example.app.SecondActivity"); //或者intent.setClassName(this.getPackageName(),"com.example.app.SecondActivity"); startActivity(intent);

3）setClass / setClassName方法

Intent intent = new Intent();

intent.setClass(this, SecondActivity.class); //或者intent.setClassName(this, "com.example.app.SecondActivity"); //或者intent.setClassName(this.getPackageName(),"com.example.app.SecondActivity"); startActivity(intent);

显式Intent通过Component可以直接设置需要调用的Activity类，可以唯一确定一个Activity，意图特别明确，所以是显式的。设置这个类的方式可以是Class对象（如SecondActivity.class），也可以是包名加类名的字符串（如"com.example.app.SecondActivity"）。

作者：翻译不了的声响  
链接：https://www.jianshu.com/p/67d99a82509b  
来源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

* **隐式Intent**  
  隐式，不明确指定启动哪个Activity，而是设置Action、Data、Category，让系统来筛选出合适的Activity。筛选是根据所有的<intent-filter>来筛选。

下面以Action为例：  
AndroidManifest.xml文件中，首先被调用的Activity要有一个带有<intent-filter>并且包含<action>的Activity，设定它能处理的Intent，并且category设为"android.intent.category.DEFAULT"。action的name是一个字符串，可以自定义，例如这里设成为"mark"：

<activity

android:name="com.example.app.SecondActivity">

<intent-filter>

<action android:name="mark"/>

<category android:name="android.intent.category.DEFAULT"/>

</intent-filter> </activity>

，在MainActivity，才可以通过这个action name找到上面的Activity。下面两种方式分别通过setAction和构造方法方法设置Action，两种方式效果相同。  
1）setAction方法

Intent intent = new Intent();

intent.setAction("mark"); startActivity(intent);

2）构造方法直接设置Action

Intent intent = new Intent("mark"); startActivity(intent);

为了防止应用程序之间互相影响，一般命名方式是**包名+Action名**，例如这里命名"mark"就很不合理了，就应该改成"com.example.app.Test"。

##### 3. Intent属性

Intent对象大致包括7大属性：**Action（动作）**、**Data（数据）**、**Category（类别）**、**Type（数据类型）**、**Component（组件）**、**Extra（扩展信息）**、**Flag（标志位）**。其中最常用的是Action属性和Data属性。

* **Action：**用来表现意图的行动  
  一个字符串变量，可以用来指定Intent要执行的动作类别。常见的action有：  
  Activity Actions：

| **类型** | **作用** |
| --- | --- |
| ACTION\_MAIN | 表示程序入口 |
| ACTION\_VIEW | 自动以最合适的方式显示Data |
| ACTION\_EDIT | 提供可以编辑的 |
| ACTION\_PICK | 选择一个一条Data，并且返回它 |
| ACTION\_DAIL | 显示Data指向的号码在拨号界面Dailer上 |
| ACTION\_CALL | 拨打Data指向的号码 |
| ACTION\_SEND | 发送Data到指定的地方 |
| ACTION\_SENDTO | 发送多组Data到指定的地方 |
| ACTION\_RUN | 运行Data，不管Data是什么 |
| ACTION\_SEARCH | 执行搜索 |
| ACTION\_WEB\_SEARCH | 执行网上搜索 |
| ACRION\_SYNC | 执同步一个Data |
| ACTION\_INSERT | 添加一个空的项到容器中 |

Broadcast Actions：

| **类型** | **作用** |
| --- | --- |
| ACTION\_TIME\_TICK | 当前时间改变，并即时发送时间，只能通过系统发送。调用格式"android.intent.action.TIME\_TICK" |
| ACTION\_TIME\_CHENGED | 设置时间。调用格式"android.intent.action.TIME\_SET" |

**Data：**表示与动作要操纵的数据  
一个URI对象是一个引用的data的表现形式，或是data的MIME类型；data的类型由Intent的action决定。

**Category：**用来表现动作的类别  
一个包含Intent额外信息的字符串，表示哪种类型的组件来处理这个Intent。任何数量的Category 描述都可以添加到Intent中，但是很多intent不需要category，下面列举一些常用的category：

| **类型** | **作用** |
| --- | --- |
| CATEGORY\_DEFAULT | 把一个组件Component设为可被implicit启动的 |
| CATEGORY\_LAUNCHER | 把一个action设置为在顶级执行。并且包含这个属性的Activity所定义的icon将取代application中定义的icon |
| CATEGORY\_BROWSABLE | 当Intent指向网络相关时，必须要添加这个类别 |
| CATEGORY\_HOME | 使Intent指向Home界面 |
| CATEGORY\_PREFERENCE | 定义的Activity是一个偏好面板Preference Panel |

**Type：**指定数据类型  
一般Intent的数据类型能够根据数据本身进行判定，但是通过设置这个属性，可以强制采用显式指定的类型而不再进行推导。

**Component：**目的组件  
指定Intent的目标组件名称，当指定了这个属性后，系统将跳过匹配其他属性，而直接匹配这个属性来启动对应的组件。

**Extra：**扩展信息  
Intent可以携带的额外 key-value 数据，你可以通过调用putExtra()方法设置数据，每一个 key对应一个 value数据。你也可以通过创建 Bundle对象来存储所有数据，然后通过调用putExtras()方法来设置数据。

| **类型** | **作用** |
| --- | --- |
| EXTRA\_BCC | 存放邮件密送人地址的字符串数组 |
| EXTRA\_CC | 存放邮件抄送人地址的字符串数组 |
| EXTRA\_EMAIL | 存放邮件地址的字符串数组 |
| EXTRA\_SUBJECT | 存放邮件主题字符串 |
| EXTRA\_TEXT | 存放邮件内容 |
| EXTRA\_KEY\_EVENT | 以KeyEvent对象方式存放触发Intent 的按键 |
| EXTRA\_PHONE\_ NUMBER | 存放调用ACTION\_CALL 时的电话号码 |

* **Flag：**期望这个意图的运行模式  
  用来指示系统如何启动一个Activity，可以通过setFlags()或者addFlags()可以把标签flag用在Intent中。

| **类型** | **作用** |
| --- | --- |
| FLAG\_ACTIVITY\_CLEAR\_TOP | 相当于SingleTask |
| FLAGE\_ACTIVITY\_SINGLE\_TOP | 相当于SingleTop |
| FLAG\_ACTIVITY\_NEW\_TASK | 类似于SingleInstance |
| FLAG\_ACTIVITY\_NO\_HISTORY | 当离开该Activity后，该Activity将被从任务栈中移除 |

##### 4. Intent用法

* 调用拨号程序

// 调用拨打电话，给10010拨打电话

Uri uri = Uri.parse("tel:10010");

Intent intent = new Intent(Intent.ACTION\_DIAL, uri);startActivity(intent);

// 直接拨打电话，需要加上权限 <uses-permission id="android.permission.CALL\_PHONE" />

Uri callUri = Uri.parse("tel:10010");

Intent intent = new Intent(Intent.ACTION\_CALL, callUri);

* 发送短信或彩信

// 给10010发送内容为“Hello”的短信

Uri uri = Uri.parse("smsto:10010");

Intent intent = new Intent(Intent.ACTION\_SENDTO, uri);

intent.putExtra("sms\_body", "Hello");startActivity(intent);

// 发送彩信（相当于发送带附件的短信）

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra("sms\_body", "Hello");

Uri uri = Uri.parse("content://media/external/images/media/23");

intent.putExtra(Intent.EXTRA\_STREAM, uri);

intent.setType("image/png");startActivity(intent);

* 通过浏览器打开网页

// 打开百度主页

Uri uri = Uri.parse("https://www.baidu.com");

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);startActivity(intent);

* 发送电子邮件

// 给someone@domain.com发邮件

Uri uri = Uri.parse("mailto:someone@domain.com");

Intent intent = new Intent(Intent.ACTION\_SENDTO, uri);startActivity(intent);

// 给someone@domain.com发邮件发送内容为“Hello”的邮件

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra(Intent.EXTRA\_EMAIL, "someone@domain.com");

intent.putExtra(Intent.EXTRA\_SUBJECT, "Subject");

intent.putExtra(Intent.EXTRA\_TEXT, "Hello");

intent.setType("text/plain");startActivity(intent);

// 给多人发邮件

Intent intent=new Intent(Intent.ACTION\_SEND);

String[] tos = {"1@abc.com", "2@abc.com"}; // 收件人

String[] ccs = {"3@abc.com", "4@abc.com"}; // 抄送

String[] bccs = {"5@abc.com", "6@abc.com"}; // 密送

intent.putExtra(Intent.EXTRA\_EMAIL, tos);

intent.putExtra(Intent.EXTRA\_CC, ccs);

intent.putExtra(Intent.EXTRA\_BCC, bccs);

intent.putExtra(Intent.EXTRA\_SUBJECT, "Subject");

intent.putExtra(Intent.EXTRA\_TEXT, "Hello");

intent.setType("message/rfc822");startActivity(intent);

* 显示地图与路径规划

// 打开Google地图中国北京位置（北纬39.9，东经116.3）

Uri uri = Uri.parse("geo:39.9,116.3");

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);startActivity(intent);

// 路径规划：从北京某地（北纬39.9，东经116.3）到上海某地（北纬31.2，东经121.4）

Uri uri = Uri.parse("http://maps.google.com/maps?f=d&saddr=39.9 116.3&daddr=31.2 121.4");

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);startActivity(intent);

* 播放多媒体

Intent intent = new Intent(Intent.ACTION\_VIEW);

Uri uri = Uri.parse("file:///sdcard/foo.mp3");

intent.setDataAndType(uri, "audio/mp3");startActivity(intent);

Uri uri = Uri.withAppendedPath(MediaStore.Audio.Media.INTERNAL\_CONTENT\_URI, "1");

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);startActivity(intent);

* 选择图片

Intent intent = new Intent(Intent.ACTION\_GET\_CONTENT);

intent.setType("image/\*");startActivityForResult(intent, 2);

* 拍照

// 打开拍照程序

Intent intent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);startActivityForResult(intent, 1);

// 取出照片数据Bundle extras = intent.getExtras();Bitmap bitmap = (Bitmap) extras.get("data");

* 获取并剪切图片

// 获取并剪切图片

Intent intent = new Intent(Intent.ACTION\_GET\_CONTENT);

intent.setType("image/\*");

intent.putExtra("crop", "true"); // 开启剪切

intent.putExtra("aspectX", 1); // 剪切的宽高比为1：2

intent.putExtra("aspectY", 2);

intent.putExtra("outputX", 20); // 保存图片的宽和高

intent.putExtra("outputY", 40);

intent.putExtra("output", Uri.fromFile(new File("/mnt/sdcard/temp"))); // 保存路径

intent.putExtra("outputFormat", "JPEG");// 返回格式startActivityForResult(intent, 0);

// 剪切特定图片

Intent intent = new Intent("com.android.camera.action.CROP");

intent.setClassName("com.android.camera", "com.android.camera.CropImage");

intent.setData(Uri.fromFile(new File("/mnt/sdcard/temp")));

intent.putExtra("outputX", 1); // 剪切的宽高比为1：2

intent.putExtra("outputY", 2);

intent.putExtra("aspectX", 20); // 保存图片的宽和高

intent.putExtra("aspectY", 40);

intent.putExtra("scale", true);

intent.putExtra("noFaceDetection", true);

intent.putExtra("output", Uri.parse("file:///mnt/sdcard/temp"));startActivityForResult(intent, 0);

* 打开手机应用市场

// 打开手机应用市场，直接进入该程序的详细页面

Uri uri = Uri.parse("market://details?id=" + packageName);

Intent intent = new Intent(Intent.ACTION\_VIEW, uri);startActivity(intent);

* 安装程序

String fileName = Environment.getExternalStorageDirectory() + "/myApp.apk";

Intent intent = new Intent(Intent.ACTION\_VIEW);

intent.setDataAndType(Uri.fromFile(new File(fileName)),"application/vnd.android.package-archive"); startActivity(intent);

* 卸载程序

Uri uri = Uri.parse("package:" + packageName);

Intent intent = new Intent(Intent.ACTION\_DELETE, uri);startActivity(intent);

* 进入设置界面

// 进入系统设置界面

Intent intent = new Intent(android.provider.Settings.ACTION\_SETTINGS);startActivity(intent);