

4.5.1 Intent的基本使用

分类 [Android 基础入门教程](#)

本节引言：

在上一节结束后意味着Android的四大组件我们都已经学习完毕了~，而本节我们要学习的是四大组件间的 枢纽——Intent(意图)，Android通信的桥梁，比如我们可以通过：

startActivity(Intent)/startActivityForResult(Intent)：来启动一个Activity

startService(Intent)/bindService(Intent)：来启动一个Service

sendBroadcast：发送广播到指定BroadcastReceiver

另外别忘了我们在注册四大组件时，写得很多的**Intent-Filter**哦~

好吧，话不多说，开始本节内容！另外前面我们已经用过Intent了，就不在讲述概念性的东西了~ 老规矩，官方API：[Intent](#)

1.显式Intent与隐式Intent的区别

显式Intent：通过组件名指定启动的目标组件,比如startActivity(new Intent(A.this,B.class)); 每次启动的组件只有一个~

隐式显式Intent:不指定组件名,而指定Intent的Action,Data,或Category,当我们启动组件时, 会去匹配AndroidManifest.xml相关组件的Intent-filter,逐一匹配出满足属性的组件,当不止一个满足时, 会弹出一个让我们选择启动哪个的对话框~

2.Intent的七个属性：

1) ComponentName(组件名称)

Android 基础入门教程(Q群号：153836263)

- 1.0 Android基础入门教程
 - 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
 - 1.2.1 使用Eclipse + ADT + S...
 - 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
 - 2.2.1 LinearLayout(线性布局)
 - 2.2.2 RelativeLayout(相对布...
 - 2.2.3 TableLayout(表格布局)
 - 2.2.4 FrameLayout(帧布局)
 - 2.2.5 GridLayout(网格布局)
 - 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)

**ComponentName
(组件名称)**

就是目标组件的名称,由组件所在应用程序配置文件中设置的包名+组件的全限定类名组成,这是显式的Intent,激发的组件只有一个!有setClass(),setClassName()或setComponent()方法设置使用getComponent()方法获取

```
ComponentName cn = new ComponentName(OneActivity.this,TwoActivity.class);
Intent it = new Intent();
it.setComponent(cn);
```

PS:其实上面的代码直接写成:Intent it = new Intent(OneActivity.this,TwoActivity.class);就可以了,所以上面的写法没什么必要,理解下就可以了,另外说下Intent中的两个参数,前者是Context,上下文,其实就是包名,后面的是要启动的Activity,由此就确定了包名+类名 = 全限定类名,从而确定启动的是哪个Activity!另外可以在第二个Activity调用getIntent().getComponent()方法获得该对象,然后调用getPackageName()获得包名,getClassName()获得类名

2) Action(动作)**Action(动作)**

一个普通的字符串,代表Intent要完成的一个抽象“动作”,比如发信息的权限,而具体由哪个组件来完成,Intent并不负责!就是仅仅知道会有这个动作,谁来完成就交给Intent-filter进行筛选了!

要注意:在Java中的Action与Intent-filter中的格式是不一样的:比如:
①<action android:name = "android.intent.action.CALL"/>
②intent.setAction(Intent.CALL_ACTION);

3) Category(类别)**Category(类别)**

同样是普通的字符串,Category则用于为Action提供额外的附加类别信息,两者通常结合使用,一个Intent对象只能有一个Action,但是能有多多个Category

同样在Java与Intent-filter中的格式也是不一样的
①<category android:name = "android.intent.category.DEFAULT"/>
②intent.addCategory(Intent.CATEGORY_DEFAULT);
可调用removeCategory()删除上次添加的种类,也可以用getCategories()方法获得当前对象所包含的全部种类

PS:Action是动作,Category是类别,为Action提供其他附加的信息,其实用的较多的是在intent-filter中使用,从而实现隐式的Intent,系统提供的相关的ACTION与CATEGORY可以自己查表:在sdk的docs文档目录中的:reference-->android-->content-->Intent.html当然如果看不懂英文话可以参考本文附录!

4) Data(数据) , Type(MIME类型)**Data(数据)
Type(MIME类型)**

Data通常用于向Action属性提供操作的数据,接受一个URI对象,URI的格式:scheme://host:port/path 参数依次为:协议头,主机,端口,路径
Type通常用于指定Data所指定的Uri对应的MIME类型,比如能够显示图片数据的组建不应该应用来播放音频文件,可以是自定义的MIME类型,只要符合abc/xyz格式的字符串就可以了

如果我们在Java代码中进行设置,另外这两个属性是会相互覆盖的
如果需要两个属性都有的话,就要调用setDataAndType()方法进行设置
而在AndroidManifest.xml文件中,这两个属性都是存放在data标签中的:
<data
android:mimeType = "Intent的Type属性"
android:scheme = "Data的scheme协议头"
android:host = "Data的主机号"
android:port = "Data的端口号"
android:path = "Data的路径"
android:pathPrefix = "Data的path前缀"
android:pathPattern = "Data属性的path的字符串模板" />

5) Extras(额外)**Extras(额外)**

通常用于多个Action之间的数据交换,Extras属性是一个Bundle对象,通过键值对进行数据的存储

单个变量:intent.putIntExtra(),getputExtra()
Bundle:intent.putExtras(),getExtras()

6) Flags(标记)

2.4.2 Date & Time组件(上)

2.4.3 Date & Time组件(下)

2.4.4 Adapter基础讲解

2.4.5 ListView简单实用

2.4.6 BaseAdapter优化

2.4.7 ListView的焦点问题

2.4.8 ListView之checkbox错...

2.4.9 ListView的数据更新问题

2.5.0 构建一个可复用的自定...

2.5.1 ListView Item多布局的...

2.5.2 GridView(网格视图)的...

2.5.3 Spinner(列表选项框)...

2.5.4 AutoCompleteTextVie...

2.5.5 ExpandableListView(...

2.5.6 ViewPager(翻转视图)...

2.5.7 Toast(吐司)的基本使用

2.5.8 Notification(状态栏通...

2.5.9 AlertDialog(对话框)详解

2.6.0 其他几种常用对话框基...

2.6.1 PopupWindow(悬浮框...

2.6.2 菜单(Menu)

2.6.3 ViewPager的简单使用

2.6.4 DrawerLayout(官方侧...

3.1.1 基于监听的事件处理机制

3.2 基于回调的事件处理机制

3.3 Handler消息传递机制浅析

3.4 TouchListener PK OnTo...

3.5 监听EditText的内容变化

3.6 响应系统设置的事件(Co...

3.7 AsyncTask异步任务

3.8 Gestures(手势)

4.1.1 Activity初学乍练

4.1.2 Activity初窥门径

4.1.3 Activity登堂入室

4.2.1 Service初涉

4.2.2 Service进阶

4.2.3 Service精通

4.3.1 BroadcastReceiver牛...

4.3.2 BroadcastReceiver庖...

4.4.1 ContentProvider初探

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概述

5.2.1 Fragment实例精讲一...

5.2.2 Fragment实例精讲一...

5.2.3 Fragment实例精讲一...

5.2.4 Fragment实例精讲一...

5.2.5 Fragment实例精讲一...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 (1)

7.3.3 Android 文件下载 (2)

7.4 Android 调用 Webservice

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScrip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

8.3.2 绘图类实战示例

Flags(标记)

表示不同来源的标记,多数用于只是Android如何启动Activity以及启动后如何对待,标记都定义在Inetnt类中,比如:FLAG_ACTIVITY_SINGLE_TOP:相当于加载模式中的singleTop模式,所有标记都是整数类型,前面Activity我们已经讲解过了~

3.显式Intent使用示例：

这个用得很多，直接就上例子了：

例子1：点击按钮返回Home界面： 运行效果图：



核心代码：

```
Intent it = new Intent();
it.setAction(Intent.ACTION_MAIN);
it.addCategory(Intent.CATEGORY_HOME);
startActivity(it);
```

例子2：点击按钮打开百度页面： 运行效果图：



核心代码：

```
Intent it = new Intent();
it.setAction(Intent.ACTION_VIEW);
it.setData(Uri.parse("http://www.baidu.com"));
startActivity(it);
```

4.隐式Intent详解

隐式Intent详解

通过比较Intent对象内容与Intent-filter过滤器来实现的主要匹配的是这三个属性: 动作,数据(URI和MIME),种类 而Extras和flags在决定哪个组件时并不起任何作用

```
<intent-filter>
  <action android:name = "com.jay.example.TEST_ACTION"/>
  <category android:name = "com.intent.category.DEFAULT"/>
  <data android:mimeType = "video/mpeg" amdroid:scheme = "http" .../>
</intent-filter>
```



1) 预定义动作的隐式Intent示例：

代码示例:点击按钮后,所有Action为VIEW的Activity被筛选出来,由用户进一步选择:

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

核心代码：

建立第二个Activity的布局,与对应的Activity,在第一个Activity的按钮点击事件中添加一下代码:

```
Intent it = new Intent();  
it.setAction(Intent.ACTION_VIEW);  
startActivity(it);
```

最后在第二个Activity的Intent中添加以下代码:

```
<activity android:name=".SecondActivity"  
    android:label="第二个Activity">  
    <intent-filter>  
        <action android:name="android.intent.action.VIEW"/>  
        <category android:name="android.intent.category.DEFAULT"/>  
    </intent-filter>  
</activity>
```

运行效果图：**2) 自定义动作的隐式Intent示例：**

代码示例：使用自定义的Action与category来激活另一个Activity

核心代码：建立第二个Activity的布局,与对应的Activity,在第一个Activity的按钮点击事件中添加一下代码:

```
Intent it = new Intent();  
it.setAction("my_action");  
it.addCategory("my_category");
```

```
startActivity(it);
```

最后在第二个Activity的Intent中添加以下代码:

```
<activity android:name=".SecondActivity"
    android:label="第二个Activity">
    <intent-filter>
        <action android:name="my_action"/>
        <category android:name="my_category"/>
        <category android:name="android.intent.category.DEFAULT"
    "/>
    </intent-filter>
</activity>
```

注意虽然我们自定义了一个category,但是还是要把这个默认的加上,不然会报错的:

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

5.常用系统Intent合集

大家贴下常用的系统Intent的合集吧,上面没有的欢迎提出~

```
//=====
==
//1.拨打电话
// 给移动客服10086拨打电话
Uri uri = Uri.parse("tel:10086");
Intent intent = new Intent(Intent.ACTION_DIAL, uri);
startActivity(intent);

//=====
==

//2.发送短信
// 给10086发送内容为"Hello"的短信
Uri uri = Uri.parse("smsto:10086");
Intent intent = new Intent(Intent.ACTION_SENDTO, uri);
intent.putExtra("sms_body", "Hello");
startActivity(intent);

//3.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);
```



```
//=====
==

//4.打开浏览器:
// 打开百度主页
Uri uri = Uri.parse("http://www.baidu.com");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//=====
==

//5.发送电子邮件: (阉割了Google服务的没戏!!!!)
// 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);

//=====
==

//6.显示地图:
// 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);

//=====
==

//7.路径规划
// 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);

//=====
==

//8.多媒体播放:
    Intent intent = new Intent(Intent.ACTION_VIEW);
    Uri uri = Uri.parse("file:///sdcard/foo.mp3");
    intent.setDataAndType(uri, "audio/mp3");
    startActivity(intent);

//获取SD卡下所有音频文件,然后播放第一首==
    Uri uri = Uri.withAppendedPath(MediaStore.Audio.Media.INTERNAL_
    CONTENT_URI, "1");
    Intent intent = new Intent(Intent.ACTION_VIEW, uri);
    startActivity(intent);

//=====
==

//9.打开摄像头拍照:
// 打开拍照程序
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(intent, 0);
// 取出照片数据
    Bundle extras = intent.getExtras();
    Bitmap bitmap = (Bitmap) extras.get("data");

//另一种:
//调用系统相机应用程序,并存储拍下来的照片
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    time = Calendar.getInstance().getTimeInMillis();
    intent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(new File(
    Environment
    .getExternalStorageDirectory().getAbsolutePath()+"/tucue", time
    + ".jpg")));
    startActivityForResult(intent, ACTIVITY_GET_CAMERA_IMAGE);

//=====
==

//10.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/te
    mp"))); // 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.C
ropImage");
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);

//=====
==

//11.打开Google Market
// 打开Google Market直接进入该程序的详细页面
Uri uri = Uri.parse("market://details?id=" + "com.demo.app");
Intent intent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(intent);

//=====
==

//12.进入手机设置界面:
// 进入无线网络设置界面(其它可以举一反三)
Intent intent = new Intent(android.provider.Settings.ACTION_WIR
ELESS_SETTINGS);
startActivityForResult(intent, 0);

//=====
==

//13.安装apk:
Uri installUri = Uri.fromParts("package", "xxx", null);
returnIt = new Intent(Intent.ACTION_PACKAGE_ADDED, installUri);

//=====
==

//14.卸载apk:
Uri uri = Uri.fromParts("package", strPackageName, null);
Intent it = new Intent(Intent.ACTION_DELETE, uri);
startActivity(it);

//=====
==

//15.发送附件:
```

```

Intent it = new Intent(Intent.ACTION_SEND);
it.putExtra(Intent.EXTRA_SUBJECT, "The email subject text");

it.putExtra(Intent.EXTRA_STREAM, "file:///sdcard/eoe.mp3");

sendIntent.setType("audio/mp3");
startActivity(Intent.createChooser(it, "Choose Email Client"));

//=====
==

//16.进入联系人页面：
Intent intent = new Intent();
intent.setAction(Intent.ACTION_VIEW);
intent.setData(People.CONTENT_URI);
startActivity(intent);

//=====
==

//17.查看指定联系人：
Uri personUri = ContentUris.withAppendedId(People.CONTENT_URI,
info.id); //info.id联系人ID
Intent intent = new Intent();
intent.setAction(Intent.ACTION_VIEW);
intent.setData(personUri);
startActivity(intent);

//=====
==

//18.调用系统编辑添加联系人（高版本SDK有效）：
Intent it = new Intent(Intent.ACTION_INSERT_OR_EDIT);
it.setType("vnd.android.cursor.item/contact");
//it.setType(Contacts.CONTENT_ITEM_TYPE);
it.putExtra("name", "myName");
it.putExtra(android.provider.Contacts.Intents.Insert.COMPANY, "
organization");
it.putExtra(android.provider.Contacts.Intents.Insert.EMAIL, "ema
il");
it.putExtra(android.provider.Contacts.Intents.Insert.PHONE, "hom
ePhone");
it.putExtra(android.provider.Contacts.Intents.Insert.SECONDARY_
PHONE, "mobilePhone");
it.putExtra(android.provider.Contacts.Intents.Insert.TERTIARY_
PHONE, "workPhone");
it.putExtra(android.provider.Contacts.Intents.Insert.JOB_TITLE,
"title");
startActivity(it);

//=====
==

```

```
//19.调用系统编辑添加联系人（全有效）：
Intent intent = new Intent(Intent.ACTION_INSERT_OR_EDIT);
intent.setType(People.CONTENT_ITEM_TYPE);
intent.putExtra(Contacts.Intents.Insert.NAME, "My Name");
intent.putExtra(Contacts.Intents.Insert.PHONE, "+1234567890");

intent.putExtra(Contacts.Intents.Insert.PHONE_TYPE, Contacts.PhonesColumns.TYPE_MOBILE);
intent.putExtra(Contacts.Intents.Insert.EMAIL, "com@com.com");

intent.putExtra(Contacts.Intents.Insert.EMAIL_TYPE, Contacts.ContactMethodsColumns.TYPE_WORK);
startActivity(intent);

//=====
==

//20.打开另一程序
Intent i = new Intent();
ComponentName cn = new ComponentName("com.example.jay.test",

"com.example.jay.test.MainActivity");
i.setComponent(cn);
i.setAction("android.intent.action.MAIN");
startActivityForResult(i, RESULT_OK);

//=====
==

//21.打开录音机
Intent mi = new Intent(Media.RECORD_SOUND_ACTION);
startActivity(mi);

//=====
==

//22.从google搜索内容
Intent intent = new Intent();
intent.setAction(Intent.ACTION_WEB_SEARCH);
intent.putExtra(SearchManager.QUERY, "searchString")
startActivity(intent);

//=====
==
```

6.Action在哪里查？

本来想直接贴以前收集到的Intent Action的，后来想想还是算了，授之以鱼，还不如授之以渔，如果你下载了Android的文档的话，可以在下述路径：

sdk-->docs-->reference-->android--->content--->Intent.html

找到这个玩意，然后从这个Constants开始就是了：

Constants

```
public static final String ACTION_AIRPLANE_MODE_CHANGED

Added in API level 1

Broadcast Action: The user has switched the phone into or out of Airplane Mode.

    • state - A boolean value indicating whether Airplane Mode is on. 1

This is a protected intent that can only be sent by the system.

Constant Value: "android.intent.action.AIRPLANE_MODE"

public static final String ACTION_ALL_APPS
```

遇到陌生的自己来这里查即可~


本节小结：


好的，关于Intent的基本使用就到这里，下一节我们会来继续学习在日常开发中使用Intent可能会遇到的一些问题或者说需求吧，敬请期待，谢谢~



坚持一个月, 看美剧不用字幕

每天只要 5分钟, 坚持30天, 效果真的不一样!





反馈

在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号
- HTML 拾色器
- JSON 格式化工具

最新更新

- Swift 正式开源
- PHP 7 正式发布
- Shell 编程快速入门
- Shell 文件包含
- Shell 输入/输出...
- Shell printf 命令
- Shell 基本运算符

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档

关注微信



Copyright © 2013-2015 菜鸟教程 [runoob.com](http://www.runoob.com) All Rights Reserved. 备案号：闽ICP备15012807号-1

