

# Android Intent and Intent-filter SDK file

Intent 使用了有一些時候了，但是往往不知道其真正的內涵，通常都是用來作為 Activity 傳遞數據的工具，另外些時候在使用 Broadcasting 來控制今天嘗試著看了下 SDK 文檔。

前半部分是敘述 Intent 的用法，後半部分是使用 Notepad 的例子來描述 Intent 的用法。其中總結的幾張表比較好，不過想看更加細節的內容還是請閱讀 android SDK 文檔吧。

## 1. Intent 使用在 Activity()中：

[Context.startActivity\(\)](#) 或者 [Activity.startActivityForResult\(\)](#)，中當然我們也可以在 get 前一個 Activity 回傳的值使用 [Activity.setResult\(\)](#) 來回傳值

## 2. Intent 使用在 Service()中：

通過 [Context.startService\(\)](#) 來喚醒某些 service 程式，或者通過 Intent 來捆綁某些服務，例如 [Context.bindService\(\)](#)

## 3. Intent 使用在 Broadcasting()中：

在廣播中我們使用 Intent 來發送某些廣播至系統例如 [Context.sendBroadcast\(\)](#)，[Context.sendOrderedBroadcast\(\)](#)，or [Context.sendStickyBroadcast\(\)](#) 這些服務都需要使用 Intent 來開啟廣播服務

綜合以上原因，android 系統發覺某些 Activity，Service 以及 Broadcasting 都是需要 Intent 的；而且這裡不會有重複發送的信息。使用在 Broadcasting 的 Intent 只能夠為 Broadcasting 來使用，而絕不會使用在 Activity 和 Service 中。一個 Intent Pass 至 Activity 中后就只能夠被 Activity 使用。絕對不會被 Service 和 Broadcasting 使用。

在下面文檔中將著重講解 Intent 和 intent-filters 組件的使用方法。

## Intent Object

一個 Intent 對象是用來捆綁信息的。

Component name（組件名）

有如下的組件：[setComponent\(\)](#)，[setClass\(\)](#)，[setClassName\(\)](#) [getComponent\(\)](#)。

## Action（動作）

Constant	Target component	Action
<code>ACTION_CALL</code>	activity	Initiate a phone call.
<code>ACTION_EDIT</code>	activity	Display data for the user to edit.

ACTION_MAIN	activity	Start up as the initial activity of a task, with no data input and no returned output.
ACTION_SYNC	activity	Synchronize data on a server with data on the mobile device.
ACTION_BATTERY_LOW	broadcast receiver	A warning that the battery is low.
ACTION_HEADSET_PLUG	broadcast receiver	A headset has been plugged into the device, or unplugged from it.
ACTION_SCREEN_ON	broadcast receiver	The screen has been turned on.
ACTION_TIMEZONE_CHANGED	broadcast receiver	The setting for the time zone has changed.

以上 Intent 的 Action 可以使用 `setAction()` 和 `getAction()` 來實現。

## Data/type

Intent 依靠 action 來判斷行為模式，然後需要依靠傳遞 data 來實現功能，而 android 的 DATA 通常情況下都使用 Uri 方式來傳遞參數。例如：

之前使用 ACTION\_EDIT 來進行編輯，這些數據將對 URI 傳送過來的數據進行編輯。如果是 ACTION\_CALL 那么 Uri 中包含了 tel: 的內容，那麼將會撥打電話，如果是 ACTION\_VIEW，只要在 Uri 中跟了 http: 的內容，那麼將會開啟這個網址。

因此 `setData()` 方法只能使用在 URI 參數中，`setType()` 只能使用在 MIME 類型，`setDataAndType()` 方法都可以使用在 URI 和 MIME 類型中。而讀 URI 使用的是 `getData()`，而獲取 MIME type 使用的是 `getType()` 方法。

URI 是 data，而 MIME 是 media type 的簡稱，意味媒體的類型。

MIME 的英文全称是 "Multipurpose Internet Mail Extensions" 多功能 Internet 邮件扩充服务，它是一种多用途网际邮件扩充协议。

具體的內容見：<http://baike.baidu.com/view/160611.html?wtp=tt#2>

## Category (種類)

一個 Intent 中可以放置多個 Category，Category 的具體內同如下：

Constant	Meaning
CATEGORY_BROWSABLE	The target activity can be safely invoked by the browser to display data referenced by a link — for example, an image or an e-mail message.
CATEGORY_GADGET	The activity can be embedded inside of another activity that hosts gadgets.

CATEGORY_HOME	The activity displays the home screen, the first screen the user sees when the device is turned on or when the HOME key is pressed.
CATEGORY_LAUNCHER	The activity can be the initial activity of a task and is listed in the top-level application launcher.
CATEGORY_PREFERENCE	The target activity is a preference panel.

## Category

有如下方法可以操作：

通過使用 [addCategory\(\)](#) 來增加一個 category Intent 的對象

使用 [removeCategory\(\)](#) 刪除一個 Category

使用 [getCategories\(\)](#) 來獲取一個 Category

## Extras

通常情況下我們使用 Extras 對象來傳遞參數，例如把一個 Activity 中的 value 傳遞到下一個 Activity 中，通常的使用是把 Bundle 一起聯用，首先先把數據放置到 bundle 中然後使用 [putExtras\(\)](#) and [getExtras\(\)](#) 方法來傳遞數據和獲取數據，當然這種方法也可以使用在 [Activity.startActivityForResult\(\)](#) 之中。

## Intent Resolution（Intent 的細節）

Intent 可以分成連部分來討論：

Explicit intent，指向組件的名稱，例如一些 Activity 和 Service 中

Implicit intents，指向沒有名稱的目標，某些域的組件是看不見的，因此需要這些內部 Intent 來啟動這些組件

通常情況下 Intent 的使用一般不需要 intent-filter 的聲明，但是如果是一個特俗操作或者聲明一些廣播操作則需要在 intent-filter 中聲明。

當一個 intent 被 intent-filter 所測試的時候通常情況下有以下三種：

action

data (both URI and data type)

category

在一個 apk 中會存在多個 Activity，service，broadcasting 等操作，因此需要使用 Intent-filter 來進行一些權限的聲明。或者屏蔽某些權限的使用。

Intent-filter 的使用可以參考 `IntentFilter()` 類來進行了解，但是通常情況下我們會在 `AndroidManifest.xml` 中來聲明；使用 `<intent-filter>` 元素來進行，但是有唯一的例外就是 filter 在注冊 Broadcasting 服務的時候需要呼叫，`Context.registerReceiver()`；來聲明。

下面是 intent-filter 的某些實驗

## Action test

An [`<intent-filter>`](#) element in the manifest file lists actions as [`<action>`](#) subelements. For example:

```
<intent-filter . . . >
    <action android:name="com.example.project.SHOW_CURRENT" />
    <action android:name="com.example.project.SHOW_RECENT" />
    <action android:name="com.example.project.SHOW_PENDING" />
    . . .
</intent-filter>
```

## Category test

An [`<intent-filter>`](#) element also lists categories as subelements. For example:

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

## Data test

使用數據的測試可以獲得某些數據的聯系例如：

```
<intent-filter . . . >
    <data android:mimeType="video/mpeg" android:scheme="http" . . . />
    <data android:mimeType="audio/mpeg" android:scheme="http" . . . />
```

```
...  
</intent-filter>
```

每一個數據元素都必須使用 URI 和 data type (MIME) 來進行表示，每一個 URI 元素包含 scheme, host, port, path 等部分

scheme://host:port/path

scheme (意義為：計劃，組合)

例如：

content://com.example.project:200/folder/subfolder/etc

這裏的 scheme 為 content，主機 host 為 com.example.project，端口 port 為 200，路徑 path 為 folder/subfolder/etc

Intent-filter Intent 濾波器，篩選器

Path 路徑是唯一的，但是 scheme 不是唯一的

<data>數據元素的 type 類型在對於 URI 來說不一定是唯一的，例如文字的類型可以用 text/\* 來表示，而音頻的類型可以用 audio/\* 來表示

Data 和 type 在 URI 和 intent 中的使用準尋一下規則：

A. 一個 Intent 對象既不包含 URI 的數據類型，也不是只有通過 intent-filter。也沒有指定仍和 URI 和數據類型。

B. 一個 Intent 包含一個 URI 但不是一個數據類型(因為不能通過一個 type 來推斷出 URI) 例如像是 mailto: 以及 tel: 這些並不是實際指的數據

C. Intent 包含的 data type，和 Intent-filter 中的數據類型應當是不相同的

## Common cases

定義 MIME type 的方式

```
<data android:mimeType="image/*" />
```

定義 scheme 和 type 共同存在的方式

```
<data android:scheme="http" android:type="video/*" />
```

通常情況下一般的 apk 啟動之后需要準尋一下兩個 intent-filter

```
<intent-filter ...>
```

```
    <action android:name="code android.intent.action.MAIN" />
```

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

</intent-filter>

下面是 SDK file 中對 Intent-filter 使用發的一些介紹

1.

action: android.intent.action.MAIN

沒有數據的 activity

2.

action: android.intent.action.MAIN

category: android.intent.category.LAUNCHER

每增加一個 Activity 就增加一個 Launcher，這樣的表示發。出現在最常見的 Activity 中

3.

action: android.intent.action.VIEW

data: content://com.google.provider.NotePad/notes

讓 Activity 中顯示后面地址中 NotePad 所有的 List 數據

content://com.google.provider.NotePad/notes. 用戶可以從這個地址 get 到自己需要的 Note

4.

action: android.intent.action.PICK

data: content://com.google.provider.NotePad/notes

用戶可以從后面地址中的 Note List 中獲取數據

content://com.google.provider.NotePad/notes. 然后 action 會返回一個 URI 來指向用戶選擇的值

5.

action: android.intent.action.GET\_CONTENT

data type: vnd.android.cursor.item/vnd.google.note

通知 Activity 支持 NotePad 的單一項目

另外在 Intent-filter 和 AndroidManifest.xml 中定義自己的 Intent，讓把自己的 Activity 的 intent 暴露出來讓其他的程序調用  
目前參考這裏的資源

<http://www.eoeandroid.com/forum-viewthread-tid-53474-highlight-Intent.html>

在 AndroidManifest.xml 中定义不就可以了么？

例如定义这个：`<action android:name="android.intent.action.VIEW" />`

别人就可以使用 `Intent intent = new Intent(Intent.ACTION_VIEW);`来调到你的 Activity 了。