[在此处输入文章标题]

当我们使用Android Studio创建一个工程并生成一个Activity时，经常可以在清单文件中看到如下的代码：

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

如果我们创建第二个Activity的话，标签里面默是没有action和category的，那么这里的android.intent.action.MAIN和android.intent.category.LAUNCHER有什么作用呢？

通过查阅API文档，了解到它们的作用如下：

**android.intent.action.MAIN**：决定应用的入口Activity，也就是我们启动应用时首先显示哪一个Activity。   
**android.intent.category.LAUNCHER**：表示activity应该被列入系统的启动器(launcher)(允许用户启动它)。Launcher是安卓系统中的桌面启动器，是桌面UI的统称。

action和category都是Intent的组成部分，前者表示意图的动作，后者表示意图的类别。所以对于上面这段话，我的理解是：android.intent.action.MAIN表明了一个应用启动时首先启动这个Activity的动作。怎样启动呢？android.intent.category.LAUNCHER告诉了我们动作是什么：通过**App的启动器**（我个人理解成桌面的图标）来启动。

**准备工作**

我们可以通过搞一些“破坏”来加深理解。在此之前先做点准备工作：简单创建一个工程（应用名称我在android：label中修改为“我的应用”），里面就有了一个默认的MainActivity，它就是入口Activity。接着我们再创建一个SecondActivity，编译器会自动帮我们在清单文件中注册好。清单文件的主要内容如下：

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".SecondActivity"></activity>

</application>

为了便于识别这两个Activity，我们在布局中都给它们添加一TextView，文本内容就是它们的名字：

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context="com.lin.mr.activityheima.MainActivity">

<TextView

android:textSize="20sp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="MainActivity" />

</RelativeLayout>

SecondActivity的布局同理，这里就不贴了。

好了，下面开始搞破坏了。

**破坏一：去掉所有Activity的android.intent.action.MAIN**

<activity android:name=".MainActivity">

<intent-filter>

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

</intent-filter>

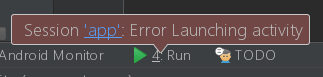
</activity>

<activity android:name=".SecondActivity"></activity>

如果我们去掉所有Activity的android.intent.action.MAIN会怎么样呢？很简单，由于android.intent.action.MAIN决定了入口Activity，去掉它之后应用就找不到入口，不知道启动哪一个Activity。如果把Activity比喻成Android应用的面孔，那么此时Android应用就没脸见人了。运行按钮旁边的选择下拉菜单就会提示当前应用有错误：

这里写图片描述

如果强行运行的话就会启动失败：



但是有一个特别的地方，打开手机设置，点击“应用管理”，发现了吗？已安装的应用中是有“我的应用”的。也就是说它能够成功安装，但是由于缺少了入口，所以桌面是看不到它的。



**破坏二：去掉所有Activity的android.intent.category.LAUNCHER**

由破坏一可知，我们必须有一个入口Activity，所以我们把android.intent.action.MAIN还给MainActivity。然后我们只需去掉MainActivity中的android.intent.category.LAUNCHER。

<activity android:name=".MainActivity">

<intent-filter>

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

</intent-filter>

</activity>

<activity android:name=".SecondActivity"></activity>

前面讲过，android.intent.category.LAUNCHER决定Activity是否列入系统的启动器，那么此时我们没有一个Activity拥有这句代码，也就是说没有一个Activity在启动列表中。结果相信你也猜到了，编译器会直接提示有错，截图就跟破坏一的一样。此时，打开手机设置中的“应用管理”，可以看到应用成功安装了，但是由于没有App的启动器，所以桌面上找不到。

**破坏三：多个Activity拥有android.intent.action.MAIN**

从破坏一和破坏二可以看到，一个应用要写正常运行是需要至少一个Activity同时拥有android.intent.action.MAIN和android.intent.category.LAUNCHER的。我们在MainActivity都加上它们，但同时我们也在SecondActivity中加入android.intent.action.MAIN：

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".SecondActivity">

<intent-filter>

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

</intent-filter>

</activity>

结果会怎么样呢？运行之后我们可以看到桌面生成了一个“我的应用”的图标，这个应用只有一个Activity，就是MainActivity。可见SecondActivity根本没有露脸。这也很好理解，它缺少了android.intent.category.LAUNCHER，不是App的启动器，点击图标启动的并不是它，我们要想看到它只能通过跳转了。





**破坏四：多个Activity同时拥有android.intent.action.MAIN和android.intent.category.LAUNCHER**

我们给SecondActivity也加上android.intent.category.LAUNCHER，这样两个Activity几乎没有差别了。同时，我们给SecondActivity再加上一句：android:label=”我的应用2”，原因后面就会知道了。

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".SecondActivity"

android:label="我的应用2">

<intent-filter>

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

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

</intent-filter>

</activity>

通过前面学习我们已经知道，这相当设置了两个入口啊，那么Android应用面对分岔路会怎么办呢？我们把破坏三中生成的“我的应用”卸载掉，再运行App。这时候神奇的事情发生了！桌面上同时生成了两个图标，一个是“我的应用”，另一个是“我的应用2”。



点击“我的应用”，我们可以看到是MainActivity，嗯，跟之前一样。那么“我的应用2”呢？点击它你会看到SecondActivity。



由此我们可以知道，当一个Android应用中有一个以上的入口Activity时，安装App时会生成相应个数的图标（你可以使用android：icon设置不同图标）。至于图标的名称，如果你没有为每一个Activity都设置的话，那么就会使用应用的名称，比如MainActivity就是这样。但是要注意，这些图标代表都是同一个应用！不信的话，你在你的手机设置中打开“应用管理”，在“已安装”中就可以看到这有一个“我的应用”，并没有“我的应用2”。



除此之外，你还可以搞最后一个破坏来验证这一点：通过拖拽图标的方式卸载应用，不管拖拽的是哪一个图标，它们最后都是一起消失的。

好了，搞了这么破坏，我对android.intent.action.MAIN和android.intent.category.LAUNCHER有了自己的理解，不知道你是怎么理解的呢？