[](https://bss.csdn.net/m/topic/blog_star2018/index?utm_source=bloghudong)

[](https://bss.csdn.net/m/topic/blog_star2018/index?utm_source=bloghudong)

关闭

原

# Android N Settings模块与Android M Settings模块差异

2016年11月03日 21:16:15 [天木楠](https://me.csdn.net/niexu19900104) 阅读数：4109

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/niexu19900104/article/details/53026132

Android N Settings 与之前的Android版本Settings模块在UI上右较大的改变，最直观的差异在于Android N 设置界面增加了向右滑动的抽屉效果。

先回想下android M设置的主界面时怎么显示和交互的？

android M设置的主界面是定义在res/xml/dashboard\_categories.xml这个文件中，其SettingsActivity.java继承与Activity,在其onCreate方法中，通过mIsShowingDashboard变量，控制加载不同的布局R.layout.settings\_main\_dashboard或者 R.layout.settings\_main\_prefs（Settings的主界面或者二级三级菜单），mIsShowingDashboard Boolear类型的变量是直接判断当前要启动的Activity的类是不是Settings.class，因为Settings这个类才是Settings的入口类。若是加载Settings的主界面，则是加载R.layout.settings\_main\_dashboard布局，这个布局里面仅仅只是一个FrameLayout的ViewContainer。

SettingsActivity.java中buildDashboardCategories（）->loadCategoriesFromResource(R.xml.dashboard\_categories, categories, this) ->updateTilesList（）去解析这个布局文件，动态添加或删除一级菜单的item，然后将主界面加载到FrameLayout的ViewContainer中，这样一级菜单就显示出来了。

对于二级菜单，在设置模块中有一个Settings.java的类，该类中定义了许多直接继承与SettingsActivity的空实现Activity,如果细心点还会发现，这些空实现的Activity在AndroidMainifest.xml中都有<meta-data/>属性，这是为什么呢？原因很简单，上面说过，既然这些Activity都是继承与SettingsActivity，那么要启动这些Activity时必然会走SettingsActivity中onCreate()方法，刚在还提到SettingsActivity中onCreate()方法是需要加载一级菜单还是加载二级菜单，取决于mIsShowingDashboard变量，而现在要确定加载的是二级菜单的具体哪个菜单？则需要用到AndroidMainifest.xml中声明该Activity时定义的<meta-data/>属性。

当然，并不是所有的设置项都是这样，例如若要在Settings中添加一个选项，跳转到其他应用中，那么也可以直接在dashboard\_categories.xml中添加相应的item，直接用intent跳转过去也时可以的。

那么，Android N中的设置主界面又时怎么显示和交互的呢？

Android N 在代码架构上和Android M最大的区别是Android N设置代码分为了两部分，一部分是在packages/apps/Settings,另一部分是在frameworks/base/packages/SettingsLib。

Android N Settings增加了向右滑动的抽屉效果，即无论在一级二级还是三级菜单界面下，只要向右滑动就可以调出Settings的主界面(事实上是无论哪一级菜单，只要改级菜单继承了SettingsActivity，那么就可以向右滑动调出设置的主界面)。接下来分析Android N Settings的启动流程。

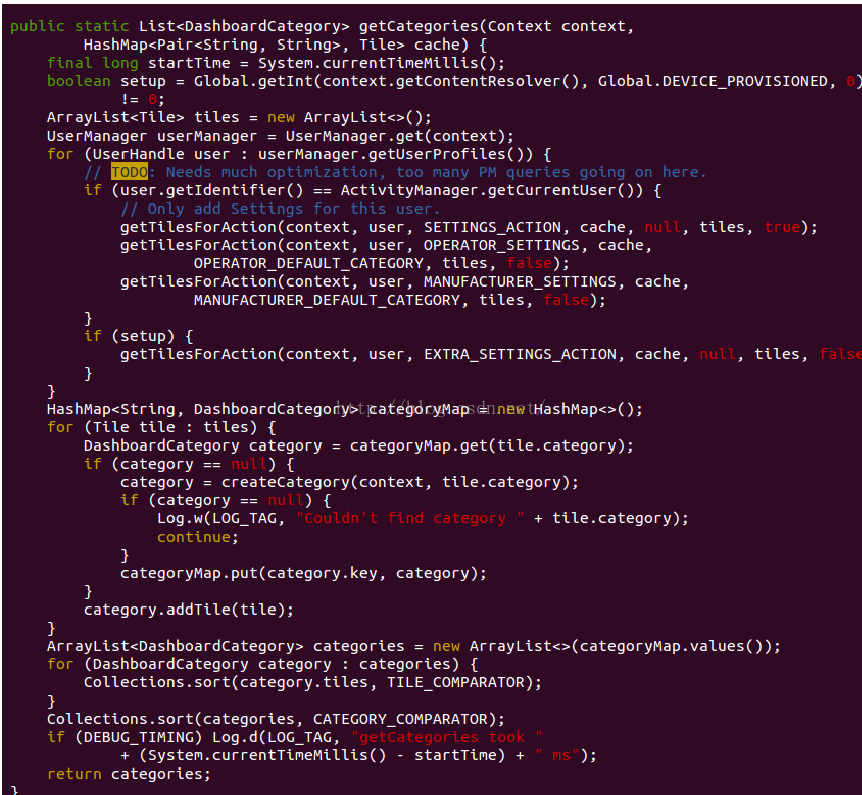
先看设置的入口类，Settings.java,该类和之前一样，没什么变化，一样继承与SettingsActivity.java，内部也定义了很多空实现的内部类，空Activity继承与SettingsActivity。那么应该和android M一样，真正的入口还是在SettingsActivity中。直接看SettingsActivity类，发现SettingsActivity不再是继承与Activity类，而是继承与rameworks/base/packages/SettingsLib下面的SettingsDrawerActivity，看一下SettingsDrawerActivity是什么鬼？？

SettingsDrawerActivity从名字来看，好像和实现滑动效果有关，到该类一看究竟，发现它加载的布局时settings\_with\_drawer.xml，这个布局中用到android.support.v4.widget.DrawerLayout，DrawerLayout 布局就具有左右滑动的抽屉效果。在看settings\_with\_drawer.xml布局中有一个listView控件，猜想该控件就是用来显示抽屉布局中的设置的主界面（后面在分析）。

继续看SettingsActivity的onCreate方法，其加载布局的地方和之前没什么区别，可是却发现有区别的地方在从DashboardSummary.class中的获取要加载item的数据部分，DashboardSummary.java的rebuildUI()中，调用了SettingsDrawerActivity的getDashboardCategories()获取数据。也就是说在设置中主界面上要显示那些选项也是rameworks/base/packages/SettingsLib中的SettingsDrawerActivity的getDashboardCategories()获取来的，而不是像Android M一样将布局定义在res/xml/dashboard\_categories.xml，现在在Settings模块下面已经没有了dashboard\_categories.xml文件，那么数据到底是怎么获取出来的呢？

现在来分析SettingsDrawerActivity类，在SettingsDrawerActivity的onCreate方法中，如上面提到，它加载的布局是settings\_with\_drawer.xml，它是外层是DrawerLayout，所以能左右滑动（可自行百度DrawerLayout用法），该布局中有listView，这个listView就是我们左右滑动看到的DrawerLayout的主界面，listView的Adapter是SettingsDrawerAdapter的实例，SettingsDrawerAdapter继承于BaseAdapter,SettingsDrawerActivity中通过getDashboardCategories()   >TileUtils.getCategories(**this**, sTileCache),获取的数据，它将数据也是封装成**DashboardCategory类型。**

**再来看 TileUtils类的getCategories()是如何返回DashboardCategory对象的List的?先看代码：**

******

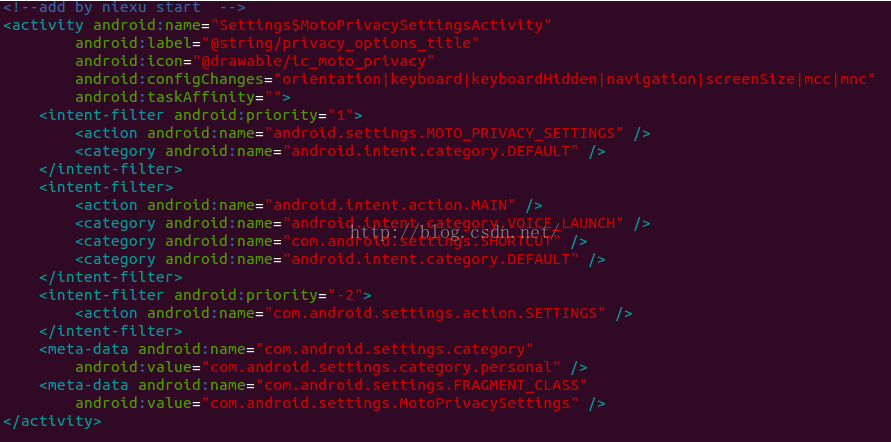
**getCategories()方法差不多可分为三步：**

**1.先调用**getTilesForAction() ->getTilesForInten（）从AndroidMainifast.xml中声明的activity信息中解析，并将解析的数据以Tile类型数据结构封装。

2，遍历存储所有Tile的集合，将通过tile.category属性，将其分类，并构建成category，并将category存储在对应的categoryMap中。

3.构建categories，并且排序，排序是按照DashboardCategory的priority排序的。

先看一下Title数据类型，它里面封装的每一个变量都是和AndroidMainifast.xml中声明Activity时<meta-data/>标签一一对应的。



Title  : 对应每个item显示的标题

icon   :对应每个Item显示的图标

summary  : 对应每个item显示的详细信息，summary

intent  :item 对应的intent

cetagory :item 属于哪一个catagory

priority  : item 的位置有关，排序会用到

extras  ：Optional additional data for use by subclasses of the activity，从启动它的类携带过来的信息

meta-data ：The metaData from the activity that defines this tile，定义的meta-data

通过这样的方法获取到了Settings主菜单和左后滑动抽屉效果需要显示的数据，然后将数据抛给SettingsDrawerAdapter，在它的getView方法中去创建对应的view，就将界面展示出来了。

总结一下：到了Android N设置中没有主界面显示的布局文件，而是直接解析AndroidManifast.xml文件，从该文件中获得并处理要显示的信息。