此文章已于 17:08:04 2018/3/29 重新发布到 Song\_UP

Android ComponentName的用法

类别 Anddroid

  ComponentName与Intent同位于Android.content包下，我们从Android官方文档中可以看到，这个类主要用来定义一个应用程序组件，例如：Activity，Service，BroadcastReceiver或者ContentProvider。  
        那么，如何用ComponentName来定义一个组件呢。  
        这是ComponentName的构造函数：ComponentName（String pkg，String cls）  
        我们知道在Android应用程序中如果要详细描述一个组件我们需要知道该组件所在的应用包名，也就是在AndroidManifest.xml文件中manifest根结点下的package=“com.XXXXX.XXXXX"，还有组件在应用程序中的完整路径名，拿Activity来说，也就是activity节点中name属性的值。因此到这里我们也就明白了可以使用ComponentName来封装一个组件的应用包名和组件的名字。  
        我们已经知道，在Android中组件之间的交流往往使用意图（Intent）来完成的，那么在Intent中有一个方法可以封装一个ComponentName，最后我们在使用意图去完成我们需要实现的功能。下面来讲一下ComponentName如何使用。

# 1.使用应用程序包名和类名启动组件

       Android 开发有时需要在一个应用中启动另一个应用，比如Launcher加载所有的已安装的程序的列表，当点击图标时可以启动另一个应用。一般我们知道了另一个应用的包名和MainActivity的名字之后便可以直接通过如下代码来启动（其中className是指完整的类名）：

**[html]**

1. Intent intent = new Intent(Intent.ACTION\_MAIN);
2. intent.addCategory(Intent.CATEGORY\_LAUNCHER);
3. ComponentName cn = new ComponentName(packageName, className);
4. intent.setComponent(cn);
5. startActivity(intent);

# 2.使用ResolveInfo启动应用程序

但是更多的时候，我们一般都不知道应用程序的启动Activity的类名，而只知道包名，我们可以通过ResolveInfo类来取得启动Acitivty的类名。

下面是实现代码：

**[html]** [view plain](https://blog.csdn.net/u012532559/article/details/52766348) [copy](https://blog.csdn.net/u012532559/article/details/52766348)

1. private void openApp(String packageName) {
2. PackageInfo pi = getPackageManager().getPackageInfo(packageName, 0);
4. Intent resolveIntent = new Intent(Intent.ACTION\_MAIN, null);
5. resolveIntent.addCategory(Intent.CATEGORY\_LAUNCHER);
6. resolveIntent.setPackage(pi.packageName);
8. List**<ResolveInfo>** apps = pm.queryIntentActivities(resolveIntent, 0);
10. ResolveInfo ri = apps.iterator().next();
11. if (ri != null ) {
12. String packageName = ri.activityInfo.packageName;
13. String className = ri.activityInfo.name;
15. Intent intent = new Intent(Intent.ACTION\_MAIN);
16. intent.addCategory(Intent.CATEGORY\_LAUNCHER);
18. ComponentName cn = new ComponentName(packageName, className);
20. intent.setComponent(cn);
21. startActivity(intent);
22. }
23. }