|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类号 |  | |  | | | 密 级 | |  |
| U D C |  | |  | | | 编 号 | | 0005 |
|  | | | | | | | | |
| **读 书 笔 记** | | | | | | | | |
|  | | | | | | | | |
| Android Studio | | | | | | | | |
|  | |  | |  |  | |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | | | | | | | | |
| By key.guan  May, 2017~ | | | | | | | | |

# 【声明】

文章来源：<http://www.cnblogs.com/smyhvae/p/4390905.html>

**【开发环境】**

物理机版本：Win7旗舰版（64位）Android Studio版本：https://developer.android.com/studio/install.html

# 第一次安装：

Android Studio安装完成后，第一次启动AS前，为了**避免重新下载新版本的SDK**，需要做如下操作：

AS启动前，打开安装目录，请先将bin目录的idea.properties文件中增加一行：disable.android.first.run=true就行了，避免第一次打开AS时自动重新下载SDK。mac平台的右键安装包->Show Package Contents 就找到bin目录了。

第一次打开Android Studio时，需要配置JDK和SDK：



上图中，选择“Project Structure”，弹出如下界面：（选择JDK和Android SDK的路径）



注意：

如果之前打开过低版本的工程提示convert，同意即可

当然，以后我们在使用Android Studio 的时候，也可以随时修改JDK和Android SDK的路径。选择「File–>Other Settings–>Default Sructure」，即可进行同样的修改。

**二、新建一个Android Studio工程：**

配置好JDK和Android SDK后，就可以开始新建Android项目了：





上图中，设置好工程名和路径之后，接下来可以一路next：



用户如果是第一次安装Android Studio，会出现如下窗口：







上图中，是在下载最新版的SDK（API 23），几分钟就下载好了，由于访问的是国外的网站，建议[FQ](http://honx.in/_U-mc6Oz5NGRmLX2S)。继续下一步：





上图中，点击“finish”按钮，应用即可创建完成。

用户如果是第一次安装Android Studio，会出现如下窗口：

[](http://images.cnitblog.com/blog/641601/201504/031950141073652.png)

上图中，是在下载Gradle，文件不大，大概几十兆吧，下载起来很快。由于访问的是国外的网站，建议[FQ](http://honx.in/_U-mc6Oz5NGRmLX2S)。

# Android Studio的简单设置：

## 主题修改：

可能大家会觉得软件的界面不太好看，我们可以换一下主题。选择菜单栏“File–settings–apperance–theme”，主题选择Darcula：



## 2、导入第三方主题：

系统提供的两种主题可能都不太好看，我们可以进入网站<http://color-themes.com/>来获取第三方主题，比如说Sublime主题就是我个人喜爱的：





上图中，下载下来之后，是一个jar包。那怎么导入到Android Studio呢？

别着急，回到Android Studio，选择菜单栏“ File-Import Settings”，将下载好的jar包导入即可。

## 3、代码字体修改：

选择菜单栏“File–settings–Editor–Colors&Fonts–Font”：



同样也可以修改控制台的字体：



修改完之后发现AS的一些默认字体如侧边栏的工程目录的字体并没有发生变化，如果想改的话，那还是改一下吧（我个人一般是不改的），修改AS的默认字体：



## 4、关闭更新：

如下图所示：



## 5、快捷键习惯的修改：

如果想修改成Eclipse的快捷键习惯，可以选择菜单栏”file–Settings–Keymap”：



我这里就不改了哈。

## 6、添加api文档悬浮提示：

AS默认是没有api文档悬浮提示的，只有按住【Ctrl+Q】太会出现提示。如果要添加api的自动悬浮提示，设置如下：



上图中，在红框部分打钩就行了，不过这样做对电脑的性能消耗会增加，可以不设置，根据个人习惯。

## 7、配置代码的自动提示：（新版AS默认具有代码自动提示的设置）

新版的AS默认具有代码自动补齐的功能（老版本的AS是没有的），自动补齐的设置如下：



保持上方红框部分出于打勾状态，就可以设置成代码自动补齐了。

代码提示的快捷键是在下面这个位置：



上图中，在搜索框输入“class name completion”，就看到了代码提示的默认快捷键是Ctrl+Alt+空格，如果想把这个快捷键改掉，操作如下：

[](http://images.cnitblog.com/blog/641601/201504/031950361398620.png)

上图中，右键选择红框部分，在弹出的对话框中进行修改。

注：如果你习惯了用Eclipse，但是要注意，android Studio中的【Alt+/】是另外一个快捷键：



习惯了用Eclipse的人可以将上图红框部分的两个快捷键交换一下。

**配置代码提示的大小写区分：**

AS默认的代码提示是**大小写敏感**的，不信看一下。我们输入小写的intent，提示效果如下：

[](http://images.cnitblog.com/blog/641601/201504/031950386399190.png)

而当输入大写的Intent时，提示效果如下：（这个时候才出现了Intent类）

[](http://images.cnitblog.com/blog/641601/201504/031950399206933.png)

上面两张图可以看到，AS默认的代码提示是**大小写敏感**的。那我想让AS对大小写不敏感，该怎么弄呢？操作如下：



上图中，我们将箭头处的部分改成“none”即可。演示效果如下：

[](http://images.cnitblog.com/blog/641601/201504/031950424822017.png)

**按住Ctrl键的同时单击鼠标查看源码：（新版AS已默认具有该设置）**

如果已经成功加载sdk，那这个其实是在快捷键里面设置的：

[](http://images.cnitblog.com/blog/641601/201504/031950436391330.jpg)

上图中的蓝栏框部分的快捷键的意思就是：按住Ctrl键的同时单击鼠标就可以查看源码。

## 设置自动导包

之前在Eclipse中，我们只有每次引用一些类的时候必须要导包，而AS可以设置成自动导包。设置如下：



上图中，将红矿部分的两个位置处勾选就行了。

## 9、**显示代码行数：**

****

上图中，将红框部分打钩即可。

## 10、禁止自动打开上次的工程：

我喜欢自己选择打开哪个工程，果断禁用（这个根据个人喜好哈），设置如下：



## 11、禁止代码折叠：

Intellij IDEA默认有很多地方的代码都会自动折叠，不过我看不惯，所以取消了，设置如下：



## 12、修改注释位置，禁用“语句堆一行”：



* Comment at frist column：启用的话，注释的位置就会处于行首，否则就根据缩进来注释。我取消掉了。
* Control statement in one line：格式化代码的时候，会把些很短的语句合并成一行。我觉得这样影响代码可读性，故取消。

## 13、修改新建文件文件头：

每次建新类的话，对下面这段注释肯定很熟悉吧？

/\*\*

\* Created by smyhvae on 2015/5/7.

\*/

其实它的设置是在下面这个位置：



## 14、修改文件编码为UTF-8：

Android Studio 1.1默认的编码方式是utf-8，到了Android Studio 1.2默认的编码方式竟然是GBK，到了Android Studio 2.1，默认的部分编码方式是utf-8，我们还是统一设置为UTF-8吧，不要坑队友哦：



15.

当我们点击android sdk的某个类进去查看源码的时候，有时候会出现如下提示：Sources for 'Android API 24 Platform' not found.

http://upload-images.jianshu.io/upload_images/278588-9d7ff3224f193708.png?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240

platform not found

简单的修复方式是打开sdk manager下载需要的sources。

# 高效开发

## [TODO 用法及自定义 TODO](http://blog.csdn.net/my_truelove/article/details/72857949)

在 android Studio 中，支持通过 TODO 在代码中插入标记，**其本质就是满足特定正则的注释**。只不过 TODO 注释能够在 Android Studio 提供的 TODO 视图窗口中查看并快速定位。开发中也是这样，遇到像下面那些问题，我们应该先记下来：

* 重复代码？ 先标记上，回头抽出去；
* 算法效率堪忧？ 先标记上，回头优化；
* 逻辑不清晰？ 先标记上，回头重构。

### TODO基本使用

在注释部分直接输入todo后，通过Android Studio 右下角中有一个 TODO tab，点击即可打开 TODO 视图，来查看项目中标记的 TODO，默认显示所有的todo，通过定义筛选器（重定义正则字符串）。



### 自定义 TODO

就是重定义正则字符串，fixme也是rodo的一个细分。现在要自定义KTODO，纯手敲一个 KTODO的 TODO 也不是不可以，只不过这样效率太低了。

用通过右上角的菜单， File -> Settings -> Editor -> Live Templates 打开代码模版设置页面，找到 AndroidComments 并展开，你会发现熟悉的 fixme 和 todo，我们就仿着它们添加我们的代码模版。点中 AndroidComments，点击右上角 +，选择 Live Template：

## 插件

Findbugs

http://www.jianshu.com/p/2b8d34b2267c

## REF

[Android Studio 必备技巧：TODO 用法及自定义 TODO](http://blog.csdn.net/my_truelove/article/details/72857949)

# 快捷键

[参考](http://seniorzhai.github.io/2015/02/05/AndroidStudio%E5%BF%AB%E6%8D%B7%E9%94%AE%E6%B1%87%E6%80%BB/)：

## **详细说明**

##IDE窗口

|按键|说明|  
|:—|:—|  
|F1|帮助|：IntelliJ IDEA 的使用说明，<https://www.jetbrains.com/help/idea/2016.2/intellij-idea-editor.html>  
|Alt(Option)+F1|查找文件所在目录位置|：弹出对话框根据数字制定显示方式（在项目中显示、在文件浏览器explorer中显示等）  
~~|Alt(Option)+1|快速打开或隐藏工程面板|：~~

|Ctrl(Command)+Alt(Option)+S|打开设置对话框|：可能跟QQ的打开消息盒子热键冲突，请把QQ的另设  
|Alt(Option)+Home|跳转到导航栏|  
|Esc|光标返回编辑框|  
|Shift+Esc|光标返回编辑框,关闭无用的窗口|  
|Shift+Click|关闭标签页|  
|F12|把焦点从编辑器移到最近使用的工具窗口|  
|Ctrl(Command)+Alt(Option)+Y|同步|  
|Ctrl(Command)+Alt(Option)+S|打开设置对话框|  
|Alt(Option)+Shift+Inert|开启/关闭列选择模式|  
|Ctrl(Command)+Alt(Option)+Shift+S|打开当前项目/模块属性|  
|Alt(Option)+Shift+C|查看文件的变更历史|  
|Alt+Shift+F10|运行|  
|Alt+Shift+F9|debug运行|  
Ctrl(Command)+Alt(Option)+F12 资源管理器打开文件夹（文件全屏）

Ctrl(Command)+Shift+F12|：代码全屏切换

##编辑  
|按键|说明|  
|:—|:—|  
|Ctrl(Command)+C|复制当前行或选中的内容|

Ctrl(Command)+Shift+V 选择粘贴

Ctrl(Command)+V 粘贴当前行或选中的内容

|Ctrl(Command)+D|删除当前行或选中的内容|  
|Ctrl(Command)+X|剪切当前行或选中的内容|  
|Ctrl(Command)+Z|倒退|：push回收站  
|Ctrl(Command)+Shift+Z|向前|：pop回收站  
|Ctrl(Command)+Alt(Option)+L|格式化代码|  
~~|Ctrl(Command)+Alt(Option)+I~~|将选中的代码进行自动缩进编排|  
|Ctrl(Command)+Alt(Option)+O|优化导入的类和包|：不需要的，新版的AS自动检测了

|Alt(Option)+Enter|自动修正|  
|Alt(Option)+Insert|得到一些Intention Action，可以生成构造器、Getter、Setter、将 == 改为 equals() 等|

|Ctrl+I|实现方法  
|Ctrl(Command)+Shift+V|选最近使用的剪贴板内容并插入|  
|Ctrl(Command)+Alt(Option)+Shift+V|简单粘贴|  
|Ctrl(Command)+Shift+Insert|选最近使用的剪贴板内容并插入（同Ctrl(Command)+Shift+V）|  
|Ctrl(Command)+Enter|在当前行的上面插入新行，并移动光标到新行（此功能光标在行首时有效）|  
|Shift+Enter|在当前行的下面插入新行，并移动光标到新行|  
|Ctrl(Command)+J|自动代码|  
|Ctrl(Command)+Alt(Option)+T|把选中的代码放在 try{} 、if{} 、 else{} 里|  
|Shift+Alt(Option)+Insert|竖编辑模式|：去选择需要竖直编辑的内容，用于多行改为同样的内容  
|Ctrl(Command)+ /|注释 // |  
|Ctrl(Command)+Shift+ /|注释 /…/|  
|Ctrl(Command)+Shift+J|合并成一行|  
|F2/Shift+F2|跳转到下/上一个错误语句处|  
|Ctrl(Command)+Shift+Back|跳转到上次编辑的地方|  
~~|Ctrl(Command)+Alt(Option)+Space|类名自动完成|~~  
~~|~~Shift+Alt(Option)+Up/Down|内容向上/下移动|  
|Ctrl(Command)+Shift+Up/Down|语句向上/下移动|  
|Ctrl(Command)+Shift+U|大小写切换|  
~~|Tab|代码标签输入完成后，按 Tab，生成代码|~~  
|Ctrl(Command)+Backspace|按单词删除|  
|Ctrl(Command)+Shift+Enter|语句完成|  
|Ctrl(Command)+Alt(Option)+J|用动态模板环绕|

##文件  
|按键|说明|  
|:—|:—|  
|Ctrl(Command)+F12|显示当前文件的结构|  
|Ctrl(Command)+H|显示类继承结构图|  
|Ctrl(Command)+Q|显示注释文档|  
~~|Ctrl(Command)+P|方法参数提示|~~  
|Ctrl(Command)+U|打开当前类的父类或者实现的接口|  
|Alt(Option)+Left/Right|切换TAB|

|Alt(Option)+Up/Down|切换光标在方法的位置|  
|Ctrl+Tab|切换Tab栈|  
|F3|快速打开光标处的类或方法|  
|Ctrl(Command)+F4|关闭当前窗口|  
~~|Ctrl(Command)+Shift+W|取消选择光标所在词|~~  
|Ctrl(Command)+ - / +|折叠/展开代码|  
|Ctrl(Command)+Shift+ - / +|折叠/展开全部代码|  
|Ctrl(Command)+Shift+.|折叠/展开当前花括号中的代码|  
|Ctrl(Command)+ ] / [|跳转到代码块结束/开始处|  
|Ctrl(Command)+Shift+C|复制路径|  
|Ctrl(Command)+Alt(Option)+Shift+C|复制引用，必须选择类名|  
|Shift+F2|切换出错的地方|

|Ctrl(Command)+G|定位行列

##查找  
|按键|说明|

|  |  |
| --- | --- |
| :— | :— |

查找用法：ctrl+右键  
|Ctrl(Command)+F|在当前窗口查找文本| + |F3|查找关键字出现位置|  
|Ctrl(Command)+Shift+F|在指定环境下查找文本|  
~~|Shift+F3|向上一个关键字出现位置|~~  
|Ctrl(Command)+R|在当前窗口替换文本|  
|Ctrl(Command)+Shift+R|查找文件名|  
|Ctrl(Command)+N|查找类|  
|Ctrl(Command)+Shift+N|查找文件|  
|Ctrl(Command)+Shift+Alt(Option)+N|查找项目中的方法或类变量或者方法局部变量|  
~~|Ctrl(Command)+B|查找变量的来源|~~  
|Ctrl(Command)+Alt(Option)+B|快速打开光标处的类或方法的实现impl|  
~~|Ctrl(Command)+Shift+B|跳转到类或方法实现处|~~  
|Ctrl(Command)+E|最近打开的文件|  
~~|Alt(Option)+F3|快速查找，效果和Ctrl(Command)+F相同|~~  
|Alt(Option)+F7|查询当前元素在工程中的引用|  
|Ctrl(Command)+F7|查询当前元素在当前文件中的引用，然后按 F3 可以选择|  
|Ctrl(Command)+Alt(Option)+F7|选中查询当前元素在工程中的引用|

##重构  
|按键|说明|  
|:—|:—|  
|F5|复制|：可以复制文件  
|F6|移动|  
|Alt(Option)+Delete|安全删除|  
|Ctrl(Command)+U|转到父类|  
|Ctrl(Command)+O|重写父类的方法|  
|Ctrl(Command)+I|实现方法|  
|Ctrl(Command)+Alt(Option)+N|内联|  
|Ctrl(Command)+Alt(Option)+Shift+T|弹出重构菜单|  
|Shift+F6|重构-重命名|  
|Ctrl(Command)+Alt(Option)+M|提取代码组成方法|  
~~|Ctrl(Command)+Alt(Option)+C|将变量更改为常量|~~  
~~|Ctrl(Command)+Alt(Option)+V|定义变量引用当前对象或者方法的返回值|~~  
|Ctrl(Command)+Alt(Option)+F|将局部变量更改为类的成员变量|  
|Ctrl(Command)+Alt(Option)+P|将变量更改为方法的参数|

##调试  
|按键|说明|  
|:—|:—|  
|F8|跳到下一步|  
|Shift+F8|跳出函数、跳到下一个断点|  
|Alt(Option)+Shift+F8|强制跳出函数|  
|F7|进入代码|  
|Shift+F7|智能进入代码|  
|Alt(Option)+Shift+F7|强制进入代码|  
|Alt(Option)+F9|运行至光标处|  
|Ctrl(Command)+Alt(Option)+F9|强制运行至光标处|  
|Ctrl(Command)+F2|停止运行|  
|Alt(Option)+F8|计算变量值|

##VCS  
|按键|说明|

## 快捷键(高频使用)

### 实现接口方法：Ctrl+I

### **快速提取方法：Ctrl + Alt + M**

### 自动生成getter等：ctrl+insert

* Shift+F12 对应于 eclipse中的Ctrl+O. 这应该是开发中的王牌快键键 **查看类的方法结构**，基本每天都需要使用几千次。
* 连按两次Shift 对应于 eclipse中的Ctrl+Shift+T和Ctrl+Shift+R **快捷搜索键**，快速搜索想要定位的文件，仅次于王牌快捷键的存在。
* Ctrl+Y **删除行** 这个可能有些人并不知道，而用Ctrl+X剪切期待，还是赶紧用这个纯粹的删除行快捷键吧。
* Ctrl+Alt+L **格式化代码** 不虚赘言，金牌按键，coding完不格式化代码的程序员都是流氓。
* Alt+Shift+Up/Down **上下移动代码**，不多说，金牌收好。
* Ctrl+O **快捷复写方法** 写个生命周期方法什么的，天天用。
* Ctrl+Alt+T **生成快捷结构体** ifelse trycatch，毛毛雨啦
* Ctrl+Shift+Plus/Minus **折叠展开全部代码块**
* Ctrl+Plus/Minus **折叠展开代码块**
* Ctrl+Alt+H **查找调用的位置** 梦里寻他千百度。
* Ctrl+Shift+U **大写小转换**
* Alt+F7 **查找一个类在哪被用到了**
* Ctrl+alt+B 查看一个接口方法的实现方法，Ctrl+B查看一个接口方法的使用
* Ctrl+F4 **关闭当前页**  
  肆意补充
* Ctrl+shift+i 不离开当前文件当前类的情况下快速查看某个方法或者类的实现。
* F11 加书签 Shift+F11 显示书签 谁用谁知道
* Ctrl+shift+a 对于没有设置快捷键或者忘记快捷键的菜单或者动作（Action），可能通过输入其名字快速调用。神技！！
* ctrl+p 在调用一些方法的时候免不了会忘记或者不知道此方法需要哪些参数。ctrl+p可以显示出此方法需要的参数。必备技能之一。
* Alt+J 选择多行编辑
* 使用方法：按住Alt加鼠标左键拉框即可

设置相关：  
studio默认的代码折叠很不喜欢，每次看都不方便，我更喜欢用Shift+F12看类的结构，所以果断关闭：  
File–>Settings–>Editor–>General–>Code Folding–>One-line methods &&”Closures”(anonymous classes implementing one method,before java 8 && Generic constructor and method parameters)

## 设置快捷键

在setting-》Keymap中可以设置各个快捷键

## Ctrl+Shift+f快捷键无效的解决方式

快捷键冲突，原因是其他程序占用了快捷键，常见的ctrl+shift+f在QQ和输入法（简体繁体切换）都可能占用， 解决思路就是找到快捷键设置，修改之，[这里有个例子](http://blog.csdn.net/java_mdzy/article/details/54140747)

# 编译环境常见问题

## [插件Plugin Error找不到的问题](http://blog.csdn.net/mrzhang_happy/article/details/53199231)

发现是许多插件无法找到 ，包括sdk等。导致项目不能运行

解决办法：点击File->Settings->Plugins: 发现Plugins下面都是红色的 ，比如[**android**](http://lib.csdn.net/base/android) Support，我报错时点进去查看的时候这个插件就是报红的，然后点击后面的对勾√再重新打上的时候Android Studio就提示你重启，点击弹出的提示框中的按钮“restart”一切正常。

[参考](http://blog.csdn.net/MrZhang_happy/article/details/53199231)

## 一直停在Refreshing gradle project

* 1. 任务管理器，kill重启AS，若不work执行第二步
  2. 手动升级gradle，more see

## Failed to resolve: com.android.databinding:library:

### 描述

使用android studio1.3.1编写dataBinding数据绑定的时候，使用的是classpath “com.android.databinding:dataBinder:1.0-rc1”，出现错误Error:Failed to resolve: com.android.databinding:library:1.0-rc1  
[Open File](http://blog.csdn.net/u012296101/article/details/openFile)  
[Show in Project Structure dialog](http://blog.csdn.net/u012296101/article/details/open.dependency.in.project.structure)  
Error:Failed to resolve: com.android.databinding:adapters:1.0-rc1  
[Open File](http://blog.csdn.net/u012296101/article/details/openFile)  
[Show in Project Structure dialog](http://blog.csdn.net/u012296101/article/details/open.dependency.in.project.structure)  


### 解决办法:

更新SDK的Extras下的Android Support Repository和Android Support Library包，更新后就可以在sdk/extras/android/m2repository/com/android/databinding的文件夹中的adapter和library文件夹中发现1.0-rc1的身影了

## 2.DexException:

### 问题描述

task ‘:app:transformClassesWithDexForDebug’.  
DexException: Multiple dex files define Landroid/AccessibilityServiceInfoCompat$AccessibilityServiceInfoVersionImpl;

### 解决办法

在app的build.gradle里面添加下面这句就好了。

android {

defaultConfig {

...

multiDexEnabled **true**

}

}

## CAN’T OPEN LOCAL TERMINAL

【问题描述】

：Java,io.IOException: Couldn’t Create PTY

无法打开Android stdio的terminal

Can’t Open Local Terminal

【解决办法】

配置要使用的终端

<http://stackoverflow.com/questions/30090800/how-to-open-a-terminal-on-webstorm>

或者下面如此

## 错误代码不提示问题解决

<http://blog.csdn.net/a_running_wolf/article/details/48708459>

今天打开AS升级了一下版本，打开以后新建工程……试着随意写几行代码练练手，结果遇到一堆问题：

* 错误代码不给提示
* Context的方法也不给提示
* …..看字太累，看图吧！

错误截图1：明显的问题代码也不给提示  


错误截图2：Context的相关方法也不给提示  


错误截图3：android studio默认Activity继承AppCompatActivity，我改成继承Activity，没有给我自动导包，而且还不提示！！！  


中间找错的过程别提有多坑爹了，以为是什么设置的问题就改设置，以为快捷键没找出来…翻了N篇文，终于碰到大神@**逆觞**[Android Studio代码自动提示无效](http://blog.csdn.net/hyr83960944/article/details/40024439)，终于发现是**AS默认模式**坑了我——**AS升级后默认设置成了省电模式，设置成省电模式了的话，AS会禁掉很多辅助功能，达到省电的目的（真是到底省不省电我不知道）。所以代码自动提示、错误提示什么的也被禁掉了。**  
找到原因就好办，一步搞定——>>  
点击“File”，倒数第二项，取消Power Save Mode的选中状态  


OK！这下所有都正常了！  


顺便吐槽一下——我就不明白这个省电模式有什么\*用！这是开发环境又不是手机，99.999%都是在工作情况下使用，还用省电？难道google是在践行我天朝可持续发展观？

在此向大神致敬！感谢所有帮助过我和可能帮助到我的大神博客

## 运行JAVA程序

参考：<http://blog.csdn.net/xiaxiayige/article/details/46706949>

目的：有时候需要写点小测试程序，测试方法是否可靠，有效。需要写个java测试程序测试，Android Studio 上运行java 程序不像Eclipse那样方便  但是Android Studio上面还是可以运行java 小程序的。

步骤1  ： File —–>  new —–> new Module…   然后弹出一个框  在more module 里面选择java  Library 如图 

然后next ，编辑自己的包名类名 然后finish 如图：



然后 就可以看见我们新建的java Library 了。 如图



然后马上写个测试程序 看看能不能运行



这个时候不能像Eclipse一样 马上执行Run按钮 ，必须先配置一下

如图配置 设置好Main Class 类路径和工作目录 如果所示 设置之前必须先添加一个application 如下图配置即可



配置完成以后 ，回到主界面 选择你配置的选项，执行Run即可 如图

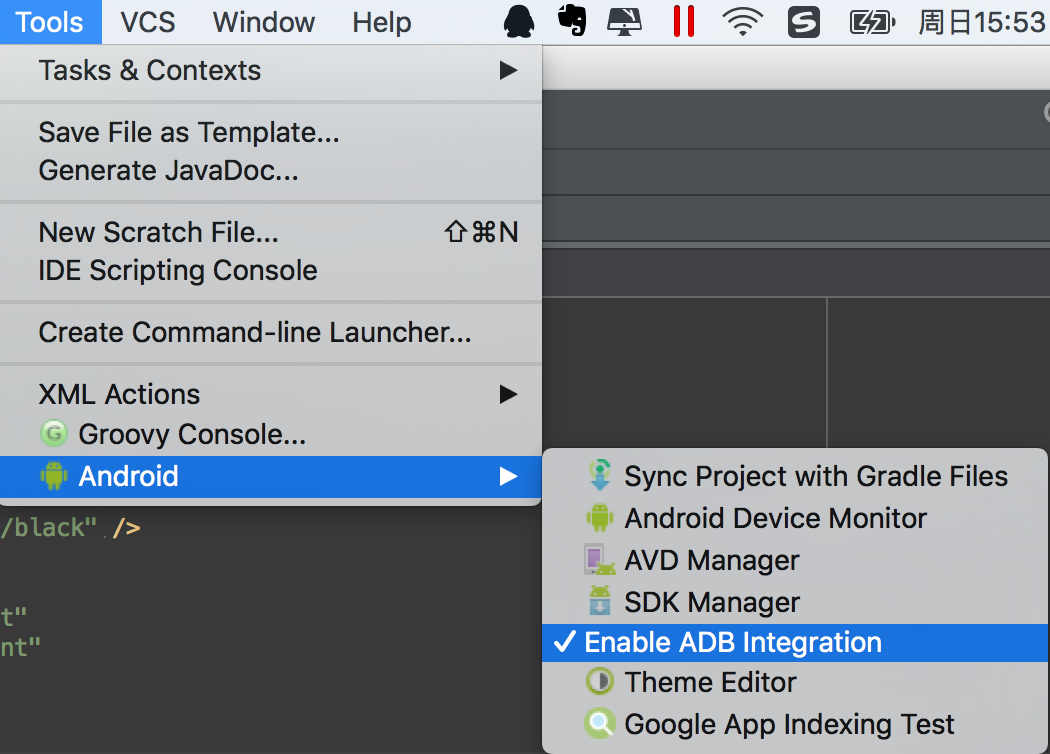


## Unable to open debugger port (localhost:8601): refused

可能两个as在debug，关闭其中一个试试。

## AS 快速debug





编译之后，找不到类，就把Enable ADB integration也关了

http://www.jianshu.com/p/3c71519032cd

# ndroid Studio编译失败提示"It is currently in use by another Gradle instance"异常处理

打开任务管理器，删另一个gradel的

# ADB使用

Android Debug Bridge

启动/关闭adb服务:adb start-server/kill-server

多设备（模拟器）指定设备号: adb devices -l; adb –s <serialNumber>

## Pm

安装/卸载/清除apk：

adb install -r XX.apk；

adb uninstall packagename

adb shell pm clear packagename 在root设备才生效！

降级安装down：adb install -r –d XX.apk

* adb shell pm list packages -f  
  查看所有已安装的应用的包名

## dumpsys

|  |  |  |
| --- | --- | --- |
|  |  |  |
| activity | 获取所用应用 | 获取所用应用:adb shell dumpsys activity  获取自己的应用:adb shell dumpsys activity | grep com.xxx.xxx.xx  获取处于栈顶的activity:adb shell dumpsys activity | grep mFocusedActivity |
| package |  |  |
| meminfo | 内存使用情况 |  |
| procstats | Memory Use Over Time |  |
| gfxinfo | Graphics State |  |

## 文件处理

adb pull <remote> <local>  
从手机复制文件出来

adb push <local> <remote>  
向手机发送文件eg. adb push foo.txt /sdcard/foo.txt

系统属性（/SYSTEM/BUILD.PROP）：adb shell getprop

日志重定向; adb logcat > bug1121.txt

CPU架构信息： /proc/cpuinfo

## 调试

### 广播am

adb shell am broadcast -a android.intent.action.test.XX --ei ei3 1：发送传整数ei3 = 1d的XX广播

adb shell am start -n packageNanme/ActivityFullName：发送启动Activity的广播

## adb无法启动

现象1：端口被占用，Unable to create Debug Bridge: Unable to start adb server: error: could not install \*smartsocket\* listener: cannot bind to 127.0.0.1:5037

现象2：unknown host service

解决办法：kill 端口占用的进程，常见的是手机助手占用了。



[Android Debugger port : Connection refused:](http://www.jianshu.com/p/96aaed625ad0) adb kill-server; adb start-server

Adb启动ok，无法发现设备

Adb是32位的，设备是64位的，请用android stdio的最新tool的adb来使用就好了。

## MIX开启 USB安装/USB调试（安全设置）

需要登录小米账号，[more](http://www.miui.com/thread-5711795-1-1.html)

然后才可以执行adb shell。

参考：

[ANDROID开发常用的ADB命令整理kg](http://www.keyguan.com/blog/2017/02/11/android%E5%BC%80%E5%8F%91%E5%B8%B8%E7%94%A8%E7%9A%84adb%E5%91%BD%E4%BB%A4%E6%95%B4%E7%90%86/)

# Linux通用命令

# 网络

netstat | grep 8088：查看网络端口，通常用来调试。

# 参考

[Keyguan](http://www.keyguan.com/blog/category/as/)

编译环境常见问题

# Task

另外有个Studio的奇技淫巧的视频，等抽时间看一看。  
<https://www.youtube.com/watch?v=Y2GC6P5hPeA>  
针对这个视频，有人的总结：  
<http://www.jianshu.com/p/c873441be31e>