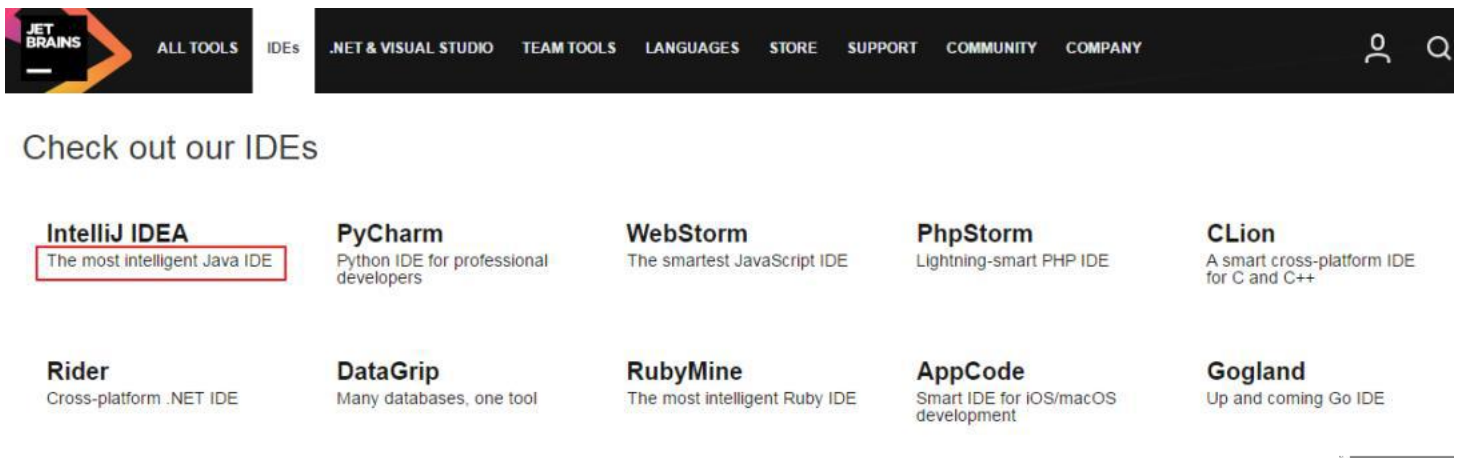


# IntelliJ IDEA的使用

## 一、IntelliJ IDEA介绍

### 1. JetBrains公司介绍

IDEA(<https://www.jetbrains.com/idea/>)是 JetBrains公司的产品，公司旗下还有其他产品，比如：WebStorm:用于开发 JavaScript、HTML5、CSS3等前端技术；Android Studio:用于开发 android(google基于 IDEA社区版进行迭代)；PhpStorm；RubyMine；PyCharm；AppCode:用于开发 OC/Swift；CLion:用于开发 C/C++等。



### 2. IntelliJ IDEA介绍

IDEA，全称 IntelliJ IDEA，是 Java语言的集成开发环境，IntelliJ在业界被公认为最好的 java开发工具之一，尤其在智能代码助手、代码自动提示、重构、J2EE支持、Ant、JUnit、CVS整合、代码审查、创新的 GUI设计等方面的功能可以说是超常的。

IntelliJ IDEA在官网上这样介绍自己：

Excel at enterprise, mobile and web development with Java, Scala and Groovy, with all the latest modern technologies and frameworks available out of the box.

简明翻译：IntelliJ IDEA主要用于支持 Java、Scala、Groovy等语言的开发工具，同时具备支持目前主流的技术和框架，擅长于企业应用、移动应用和 Web应用的开发。

### 3.IDEA的主要功能介绍

语言支持上：

安装插件后支持	SQL类	基本JVM
PHP	PostgreSQL	Java
Python	MySQL	Groovy
Ruby	Oracle	
Scala	SQL Server	
Kotlin		
Clojure		

其他支持：

支持的框架	额外支持的语言代码提示	支持的容器
Spring MVC	HTML5	Tomcat
GWT	CSS3	TomEE
Vaadin	SASS	WebLogin
Play	LESS	JBoss
Grails	JavaScript	Jetty
Web Services	CoffeeScript	WebSphere
JSF	Node.js	
Struts	ActionScript	
Hibernate		
Flex		

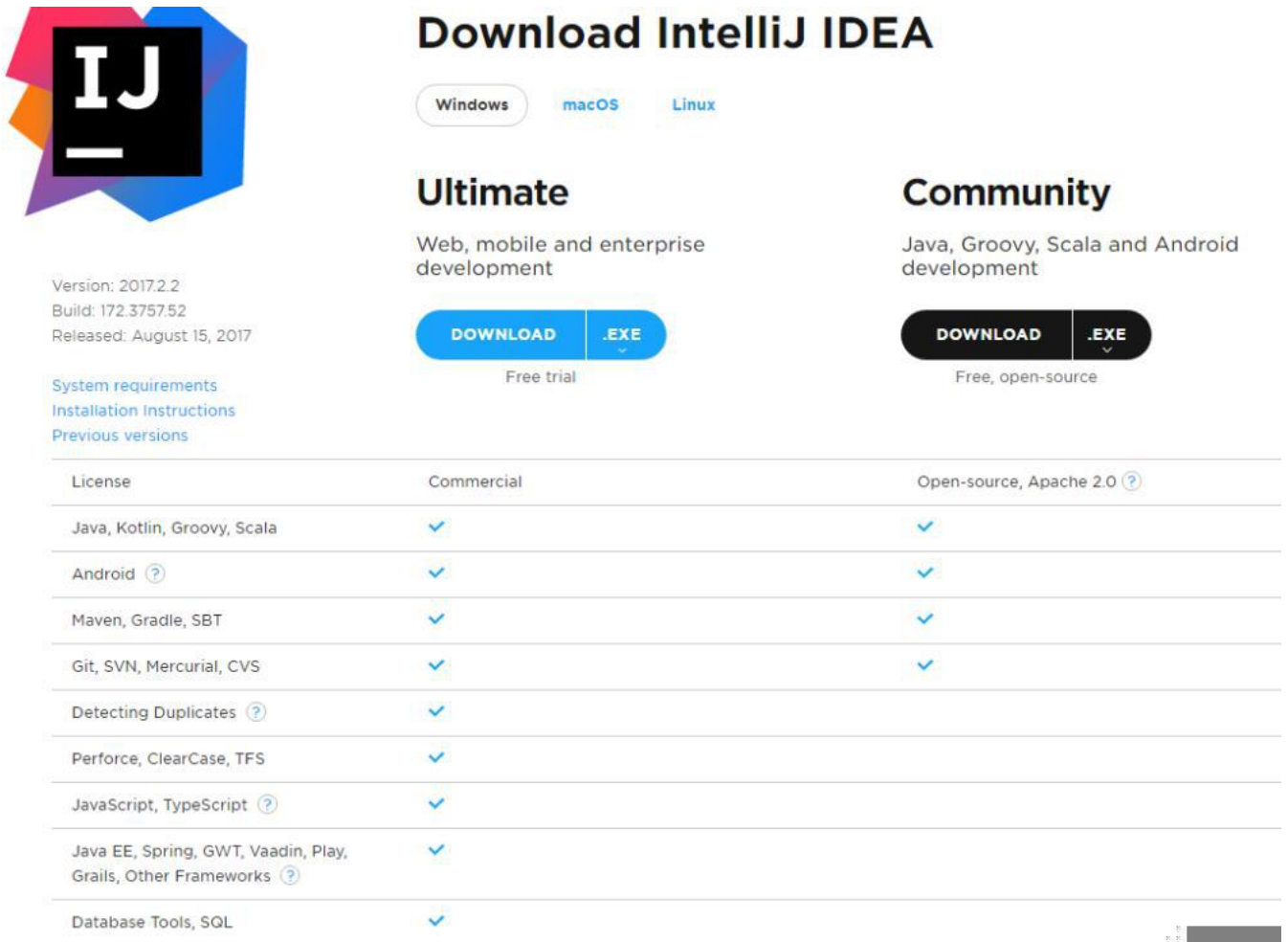
上面特性只是 IntelliJ IDEA的冰山一角，而且这个还不是 IntelliJ IDEA最重要的地方，IntelliJ IDEA最重要的特性就是人性化、智能。

## 4.IDEA的下载地址：(官网)

<https://www.jetbrains.com/idea/download/#section=windows>

IDEA不是免费的，这和 Eclipse有很大区别。

IDEA分为两个版本，旗舰版(Ultimate)和社区版(Community)



The screenshot shows the IntelliJ IDEA download page. On the left is the IntelliJ logo. Below it, version information is listed: Version: 2017.2.2, Build: 172.3757.52, Released: August 15, 2017. Links for System requirements, Installation instructions, and Previous versions are provided. The main section is titled 'Download IntelliJ IDEA' and offers three operating system options: Windows, macOS, and Linux. Two versions are highlighted: 'Ultimate' (Web, mobile and enterprise development) and 'Community' (Java, Groovy, Scala and Android development). Each version has a 'DOWNLOAD' button and a '.EXE' button. Below the download buttons, a table compares the features of the Commercial (Ultimate) and Open-source (Community) licenses.

License	Commercial	Open-source, Apache 2.0 ?
Java, Kotlin, Groovy, Scala	✓	✓
Android ?	✓	✓
Maven, Gradle, SBT	✓	✓
Git, SVN, Mercurial, CVS	✓	✓
Detecting Duplicates ?	✓	
Perforce, ClearCase, TFS	✓	
JavaScript, TypeScript ?	✓	
Java EE, Spring, GWT, Vaadin, Play, Grails, Other Frameworks ?	✓	
Database Tools, SQL	✓	

本篇内容为宋老师为大家整理的 IntelliJ IDEA的常用设置和工程创建等。

## 二、windows下安装过程

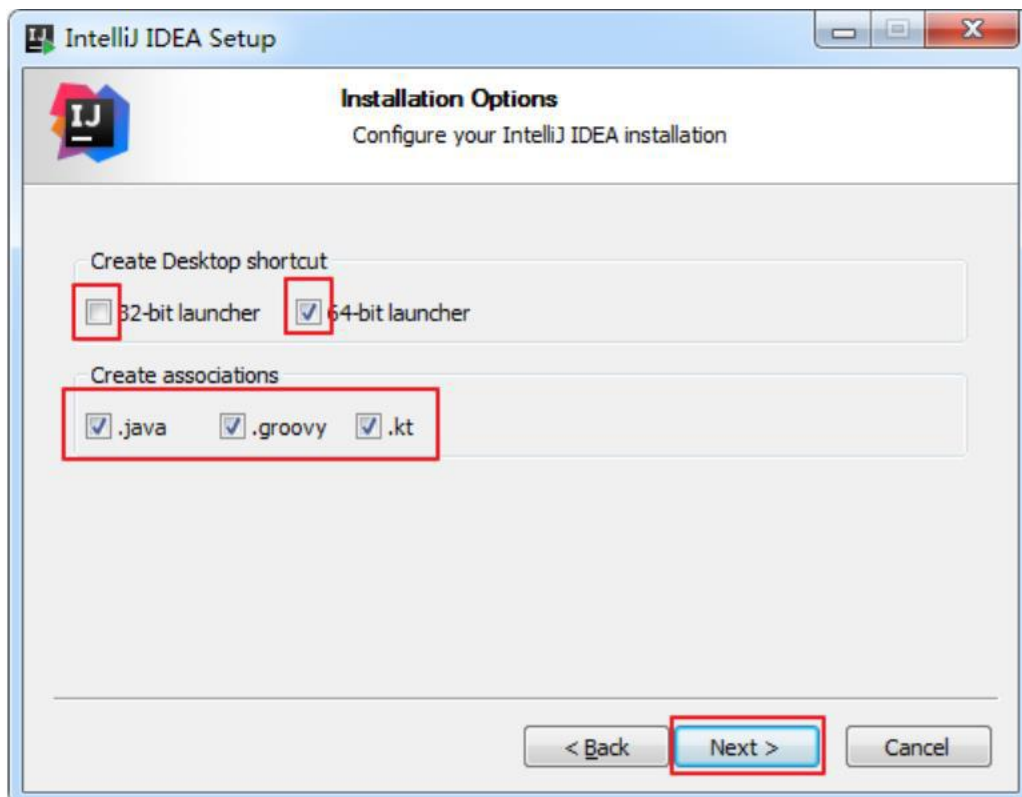
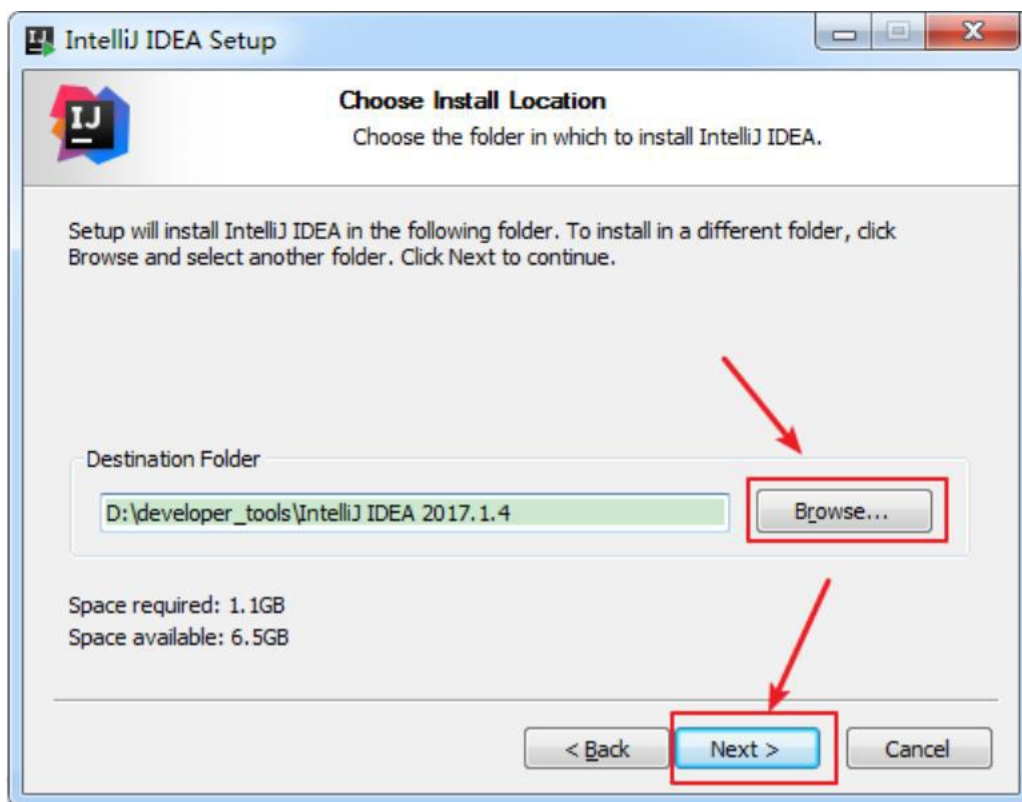
### 1.安装前的准备

注意：安装之前，确认本机已经安装了 JDK。这里如果没有安装的话，请参考提供的文档《尚硅谷\_宋红康\_JDK8的下载\_安装\_配置.pdf》进行安装配置。

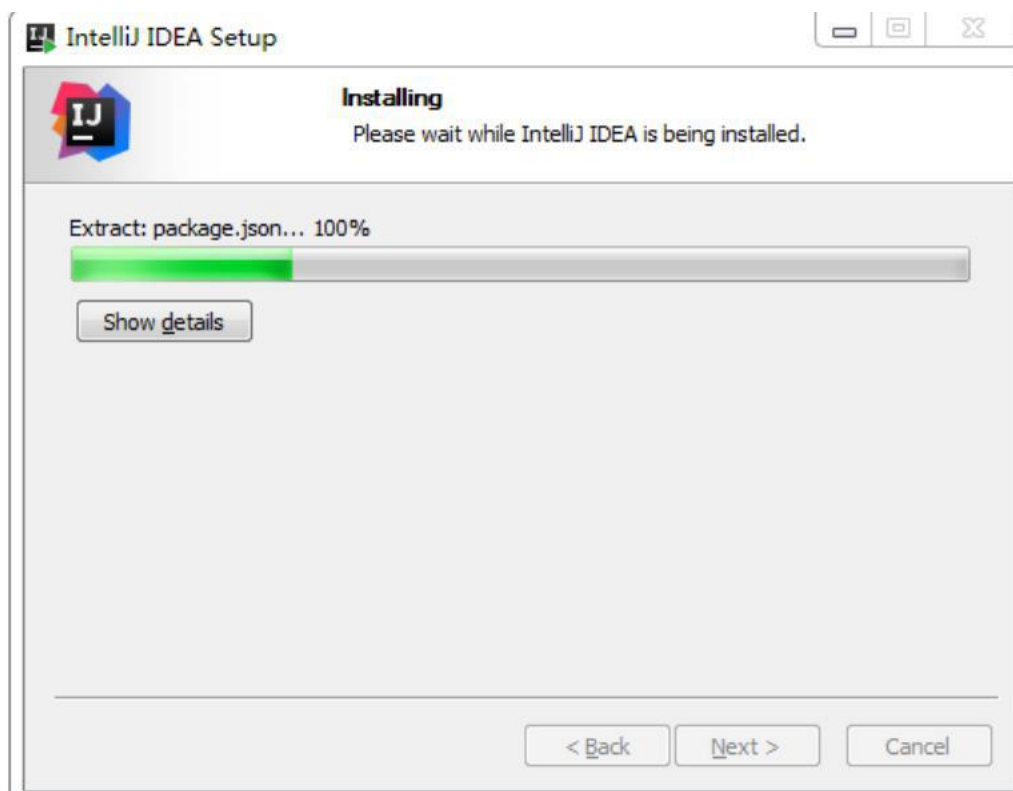
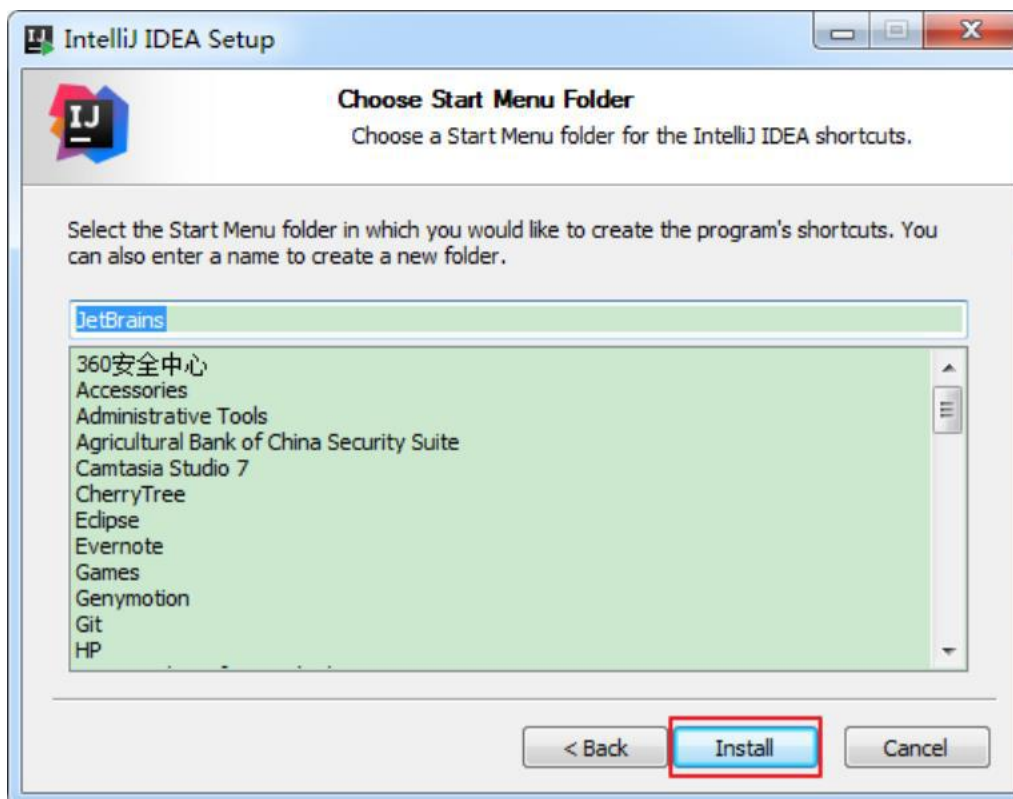
## 2.具体安装过程

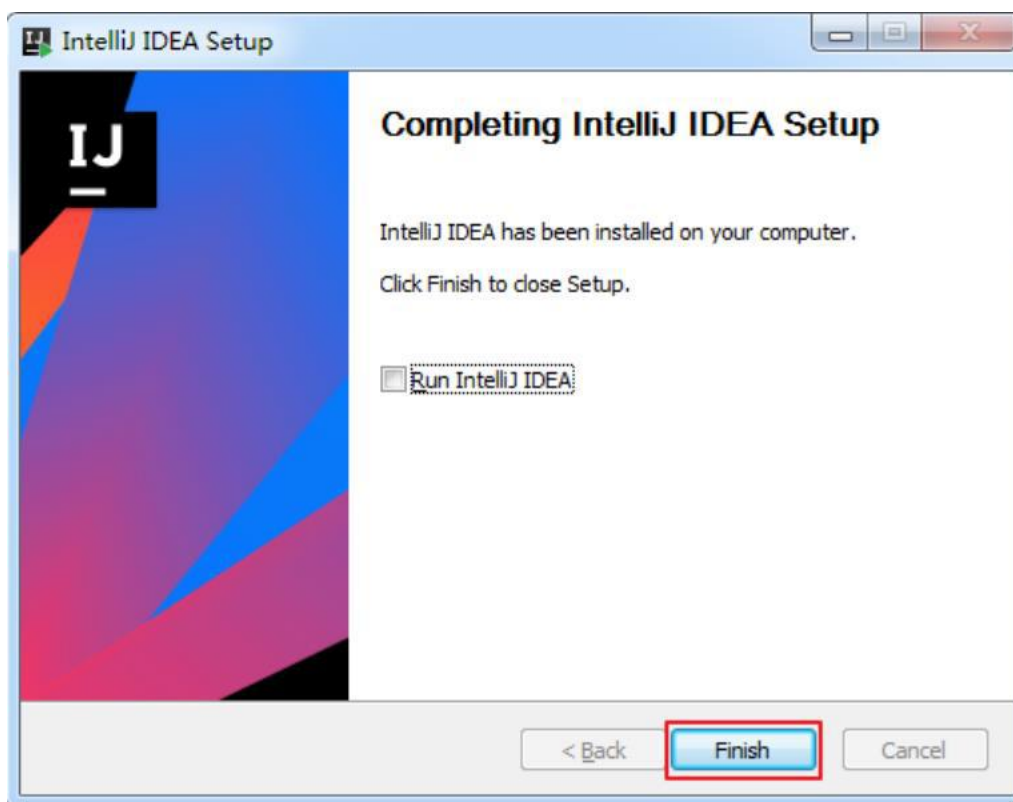
双击：





- 确认 32位版还是 64位版
- 确认是否与.java、.groovy、.kt格式文件进行关联，这里也可以选择不关联。  
(groovy:一种基于 JVM的敏捷开发语言)





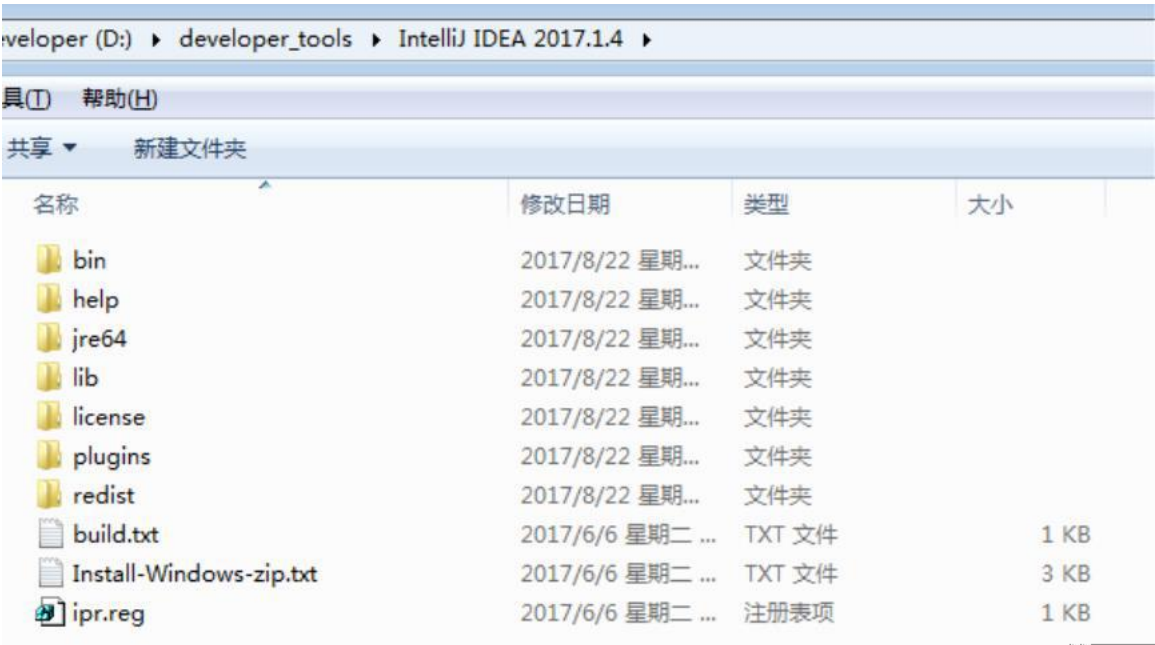
### 3. 安装总结

从安装上来看，IntelliJ IDEA对硬件的要求似乎不是很高。可是实际在开发中其实并不是这样的，因为 IntelliJ IDEA执行时会有大量的缓存、索引文件，所以如果你正在使用 Eclipse / MyEclipse，想通过 IntelliJ IDEA来解决计算机的卡、慢等问题，我这里可以直接明白地告诉你：这基本上是不可能的，本质上你应该对自己的硬件设备进行升级。

**建议配置：**内存 **8G**或以上，最好有块固态硬盘，将 **IDEA**安装在固态硬盘上，这样流畅度会加快很多。



## 4.查看安装目录结构



The screenshot shows the file explorer for the IntelliJ IDEA 2017.1.4 installation directory. The path is D:\developer\_tools\IntelliJ IDEA 2017.1.4. The table below represents the contents of this directory.

名称	修改日期	类型	大小
bin	2017/8/22 星期...	文件夹	
help	2017/8/22 星期...	文件夹	
jre64	2017/8/22 星期...	文件夹	
lib	2017/8/22 星期...	文件夹	
license	2017/8/22 星期...	文件夹	
plugins	2017/8/22 星期...	文件夹	
redist	2017/8/22 星期...	文件夹	
build.txt	2017/6/6 星期二 ...	TXT 文件	1 KB
Install-Windows-zip.txt	2017/6/6 星期二 ...	TXT 文件	3 KB
ipr.reg	2017/6/6 星期二 ...	注册表项	1 KB

bin: 容器，执行文件和启动参数等

help: 快捷键文档和其他帮助文档

jre64: 64 位java 运行环境

lib: idea 依赖的类库

license: 各插件许可

plugin: 插件

其中: bin目录下:

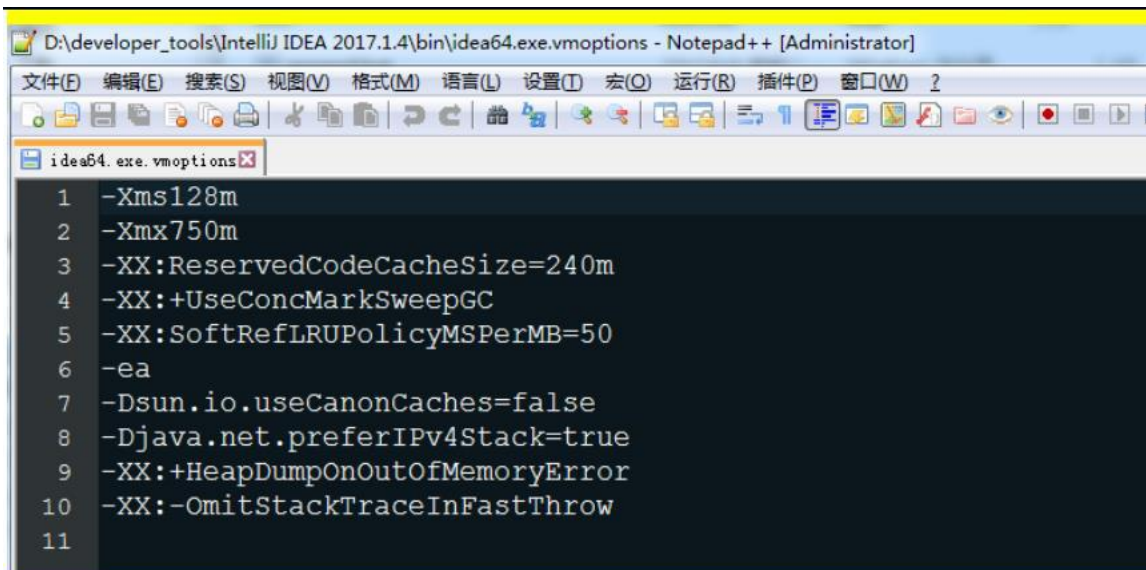


The screenshot shows the contents of the bin directory with red arrows pointing to specific files and their descriptions:

idea.bat	32位IDEA启动文件	Windows 批处理...
idea.exe	32位IDEA的VM配置文件	应用程序
idea.exe.vmoptions	IDEA的属性配置文件	VMOPTIONS 文件
idea.ico		图标
idea.properties		PROPERTIES 文件
idea64.exe	64位IDEA启动文件	应用程序
idea64.exe.vmoptions	64位IDEA的VM配置文件	VMOPTIONS 文件
IdeaWin32.dll		应用程序扩展
IdeaWin64.dll		应用程序扩展

这里以我的电脑系统(64位 windows7, 16G内存)为例, 说明一下如何调整 VM配置文件:





```
1 -Xms128m
2 -Xmx750m
3 -XX:ReservedCodeCacheSize=240m
4 -XX:+UseConcMarkSweepGC
5 -XX:SoftRefLRUPolicyMSPerMB=50
6 -ea
7 -Dsun.io.useCanonCaches=false
8 -Djava.net.preferIPv4Stack=true
9 -XX:+HeapDumpOnOutOfMemoryError
10 -XX:-OmitStackTraceInFastThrow
11
```

- 1.根据各位电脑系统的位数，选择 32位的 VM配置文件或者 64位的 VM配置文件
2. 32位操作系统内存不会超过 4G，所以没有多大空间可以调整，建议不用调整了
3. 64位操作系统中 8G内存以下的机器或是静态页面开发者是无需修改的。
4. 64位操作系统且内存大于 8G的，如果你是开发大型项目、Java项目或是 Android项目，建议进行修改，常修改的就是下面 3个参数：

-Xms128m，16 G内存的机器可尝试设置为 -Xms512m

(设置初始的内存数，增加该值可以提高 Java程序的启动速度。)

-Xmx750m，16 G内存的机器可尝试设置为 -Xmx1500m

(设置最大内存数，提高该值，可以减少内存 Garbage收集的频率，提高程序性能)

-XX:ReservedCodeCacheSize=240m，16G内存的机器可尝试设置为 -XX:ReservedCodeCacheSize=500m  
(保留代码占用的内存容量)

## 5.查看设置目录结构



对于这个设置目录有一个特性，就是你删除掉整个目录之后，重新启动 IntelliJ IDEA会再自

动帮你再生成一个全新的默认配置，所以很多时候如果你把 IntelliJ IDEA配置改坏了，没关系，删掉该目录，一切都会还原到默认。

其中，**config**目录：

**config**目录是 IntelliJ IDEA个性化配置目录，或者说是整个 IDE设置目录。也是我个人认为最重要的目录，没有之一，如果你还记得安装篇的介绍的时候，安装新版本的 IntelliJ IDEA会自动扫描硬盘上的旧配置目录，指的就是该目录。这个目录主要记录了：IDE主要配置功能、自定义的代码模板、自定义的文件模板、自定义的快捷键、Project的 tasks记录等等个性化的设置。

其中，**system**目录：

**system**目录是 IntelliJ IDEA系统文件目录，是 IntelliJ IDEA与开发项目一个桥梁目录，里面主要有：缓存、索引、容器文件输出等等，虽然不是最重要目录，但是也是最不可或缺目录之一。

## 三、启动应用后简单配置

### 1.是否导入已有设置

首次启动，会弹出如下的对话框。选择不导入已有的设置。



### 2.激活

然后根据提供的激活文档，填入：licence server的具体值即可。（需要联网）  
或者选择 Activation code，根据文档提供的激活码，同样可以激活。（不需要联网）

IntelliJ IDEA License Activation

☒ Activate   ☐ Evaluate for free   [Buy IntelliJ IDEA](#)

Activate license with:

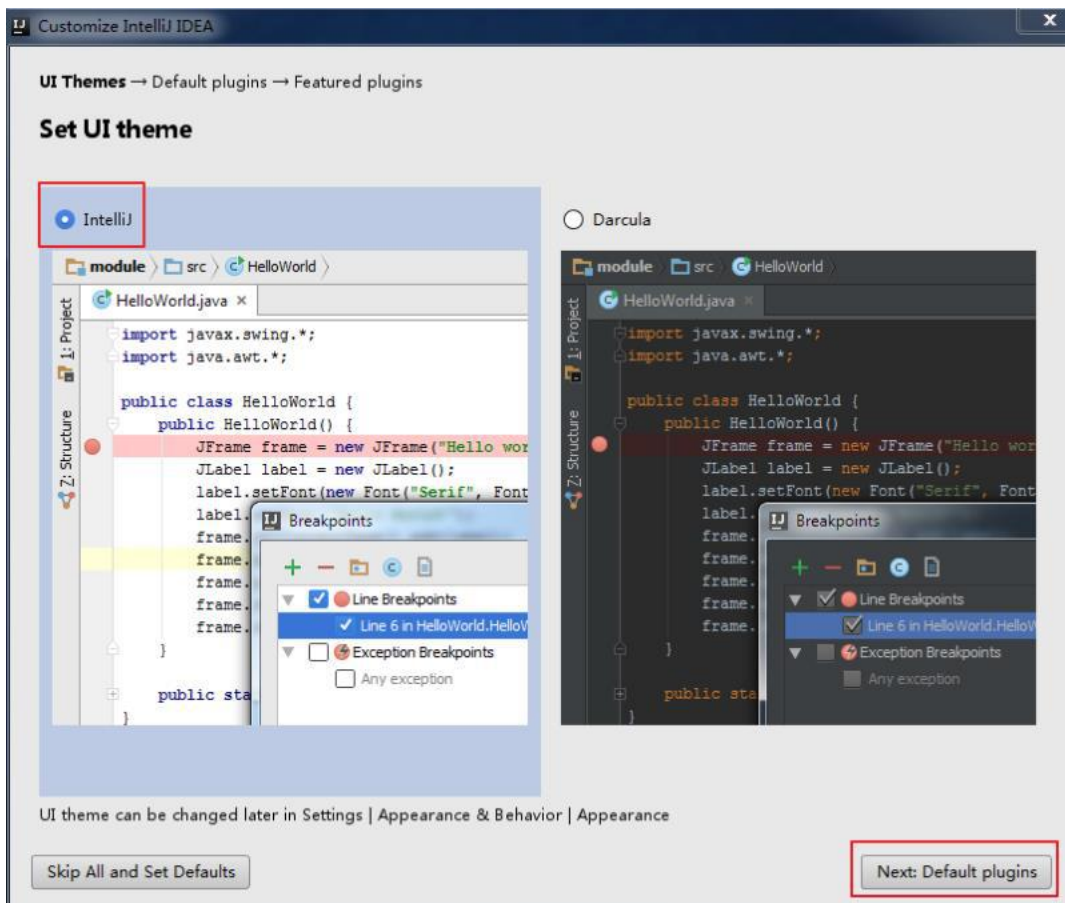
☐ JetBrains Account   ☐ Activation code   ☒ License server

License server address: [More info...](#)

[Discover server](#)

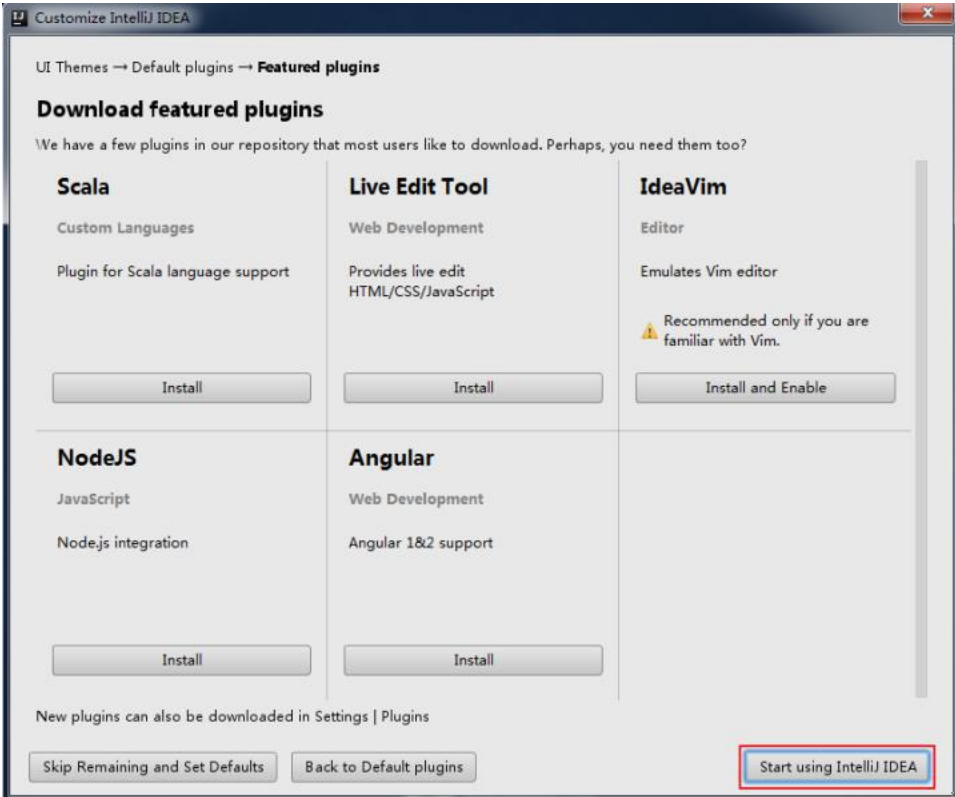
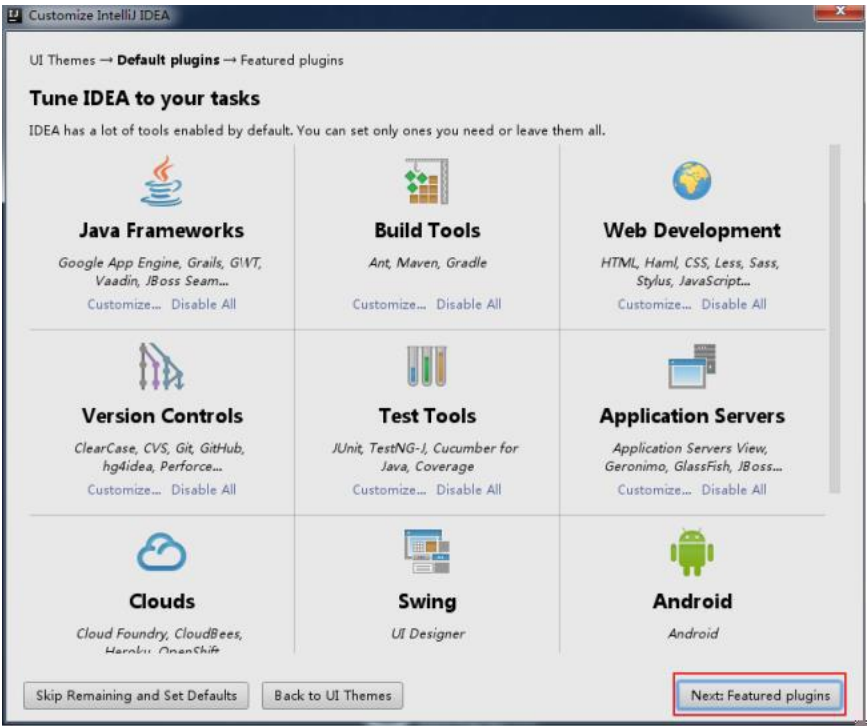
[Activate](#)   [Exit](#)

### 3. 设置主题



这里根据个人喜好，进行选择，也可以选择跳过(skip all and set defaults)。后面在 settings 里也可以再设置主题等。

## 4.其它



---

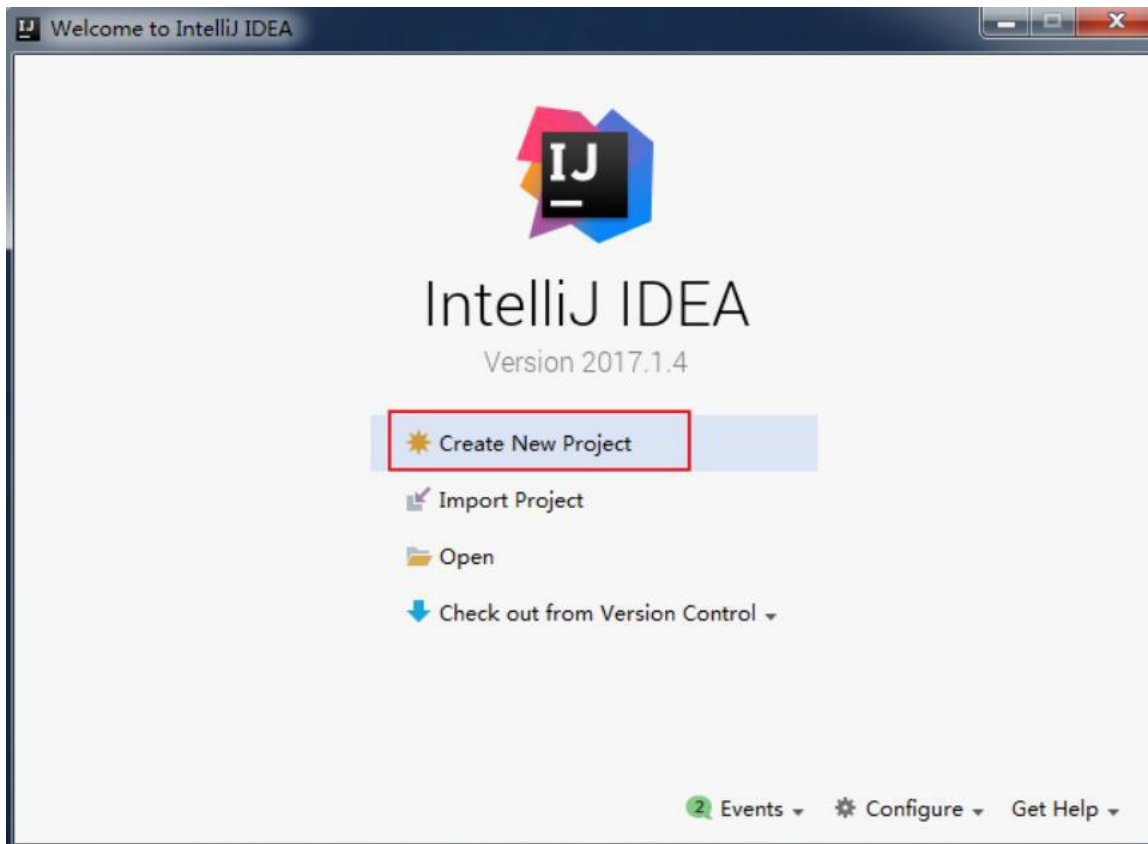
## 5.启动页面



---

## 四、创建 Java 工程

### 1. 创建工程

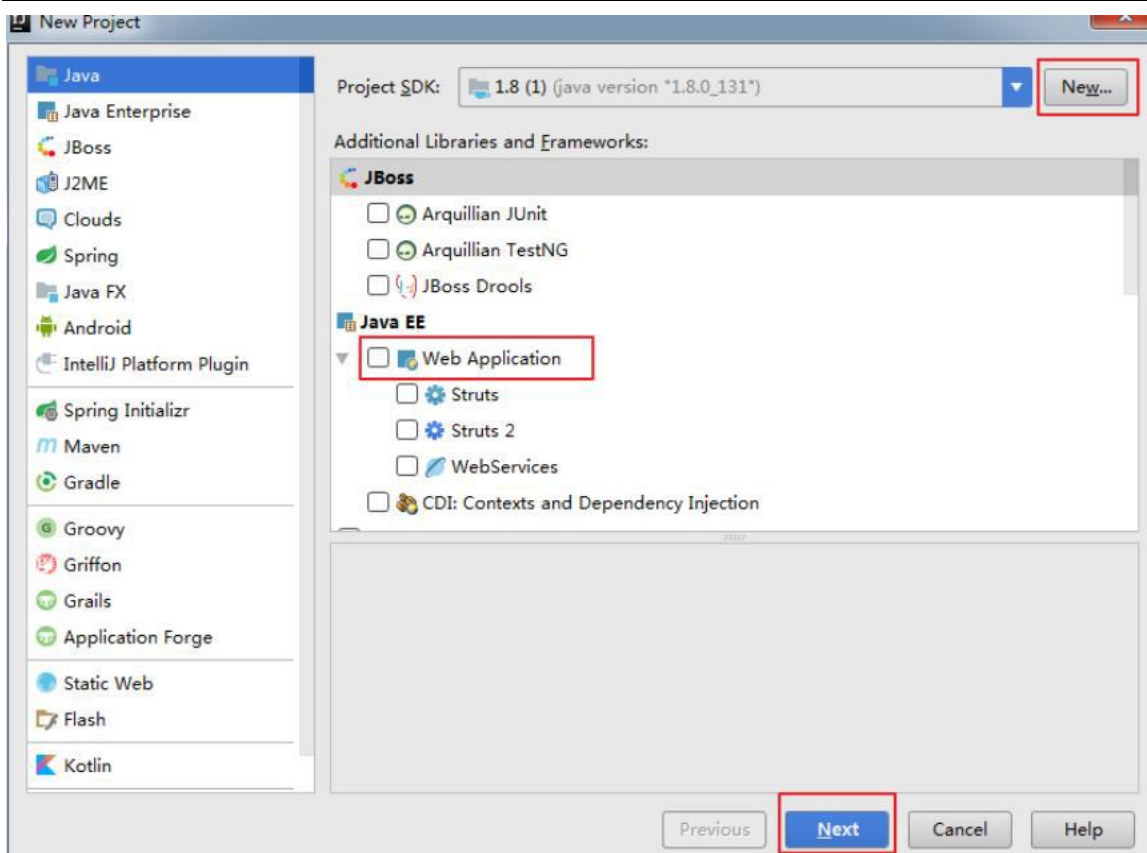


- Create New Project: 创建一个新的工程
- Import Project: 导入一个现有的工程
- Open: 打开一个已有工程，可以直接打开 Eclipse 项目，但由于两个 IDE 的项目配置不一样，所以项目还是需要配置的
- Check out from Version Control: 可以通过服务器上的项目地址 check out Github 上面项目或其他 Git 托管服务器上的项目

这里选择 Create New Project，重点说明：

IntelliJ IDEA 没有类似 Eclipse 的工作空间的概念（**Workspaces**），最大单元就是 **Project**。如果你同时观察多个项目的情况，IntelliJ IDEA 提供的解决方案是打开多个项目实例，你可以理解为开多个项目窗口。

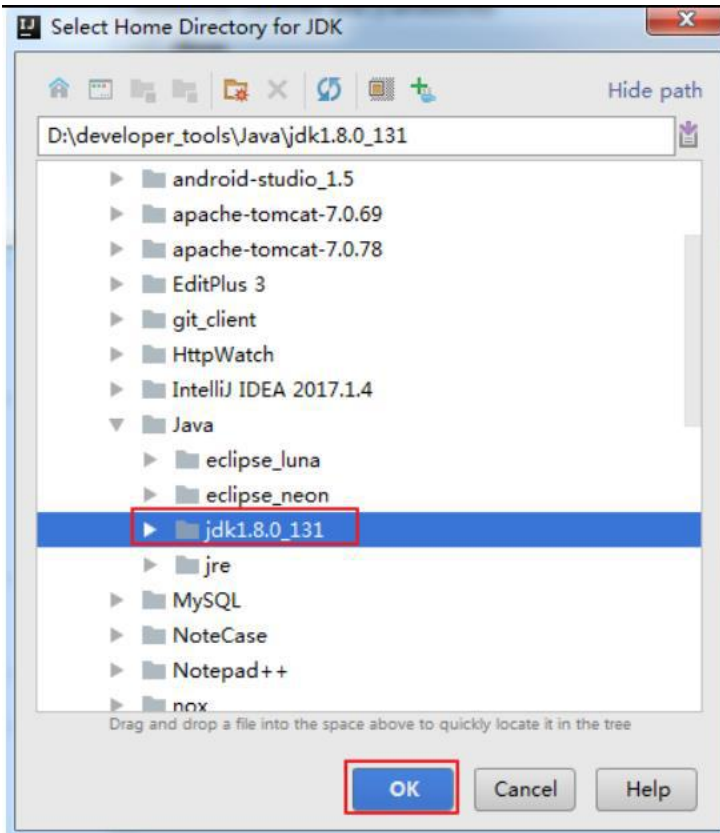




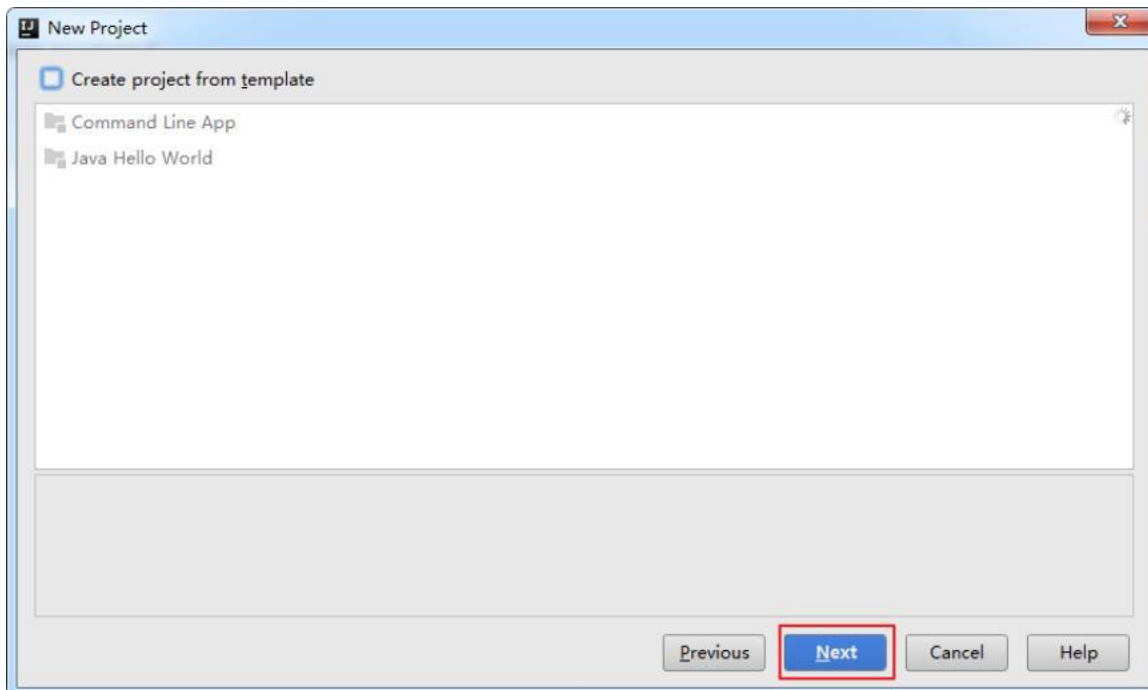
选择 JDK 作为 SDK。如果需要创建 Web 工程，则需要勾选上面的 Web Application。如果不需要创建 Web 工程的话，则不需要勾选。这里先不勾选，只是创建简单的 Java 工程。

其中，选择 New:

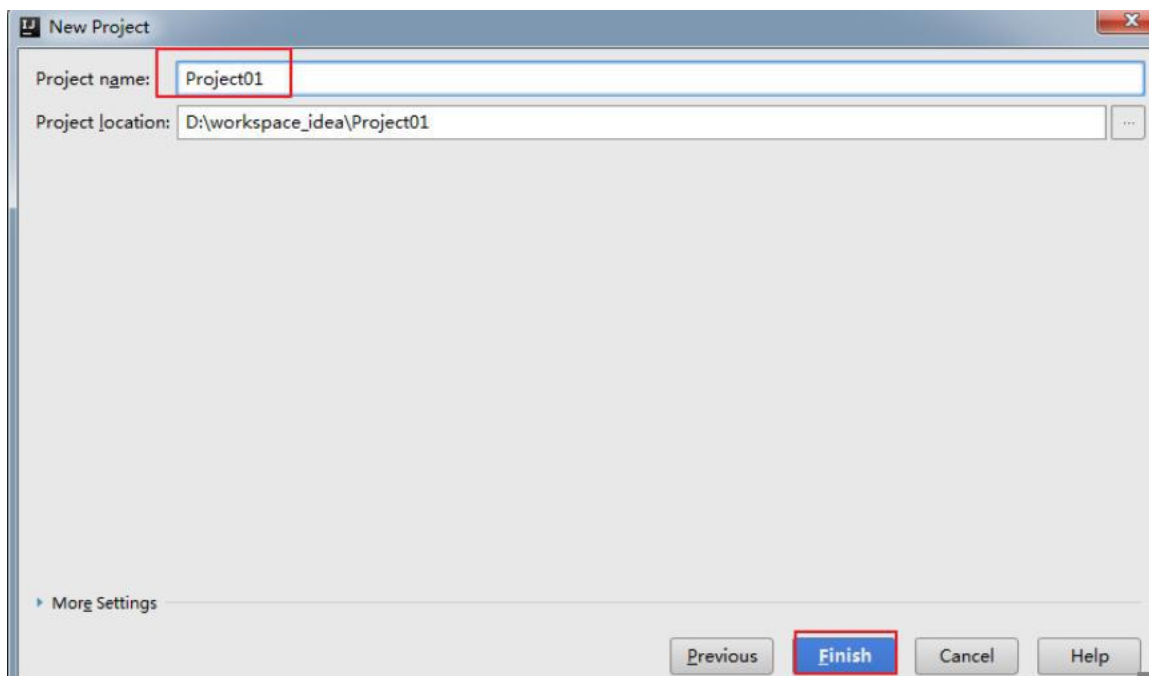
选择 jdk 的安装路径所在位置:



点击 OK以后，选择 Next:

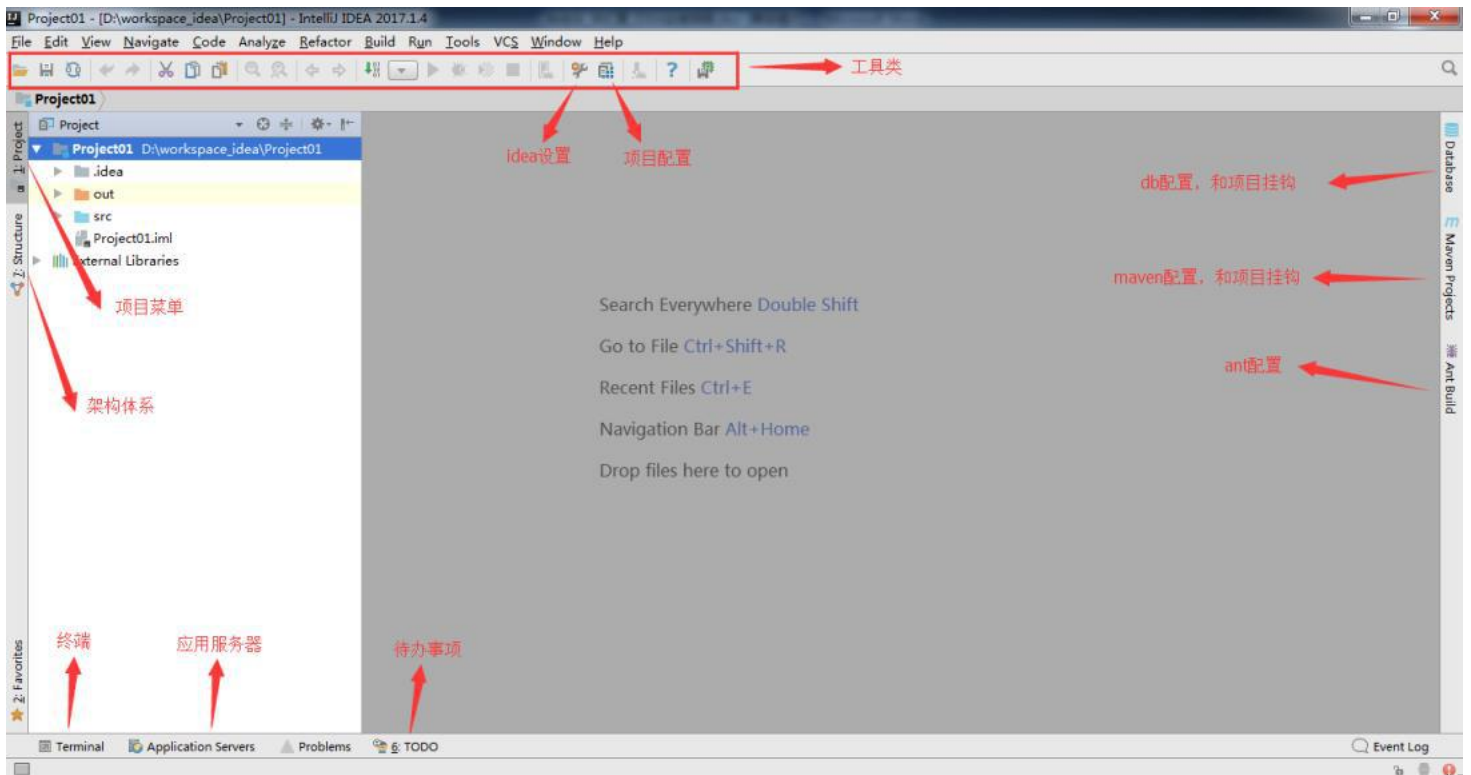


这里不用勾选。选择 Next，进入下一个页面：



给创建的工程起一个名字。

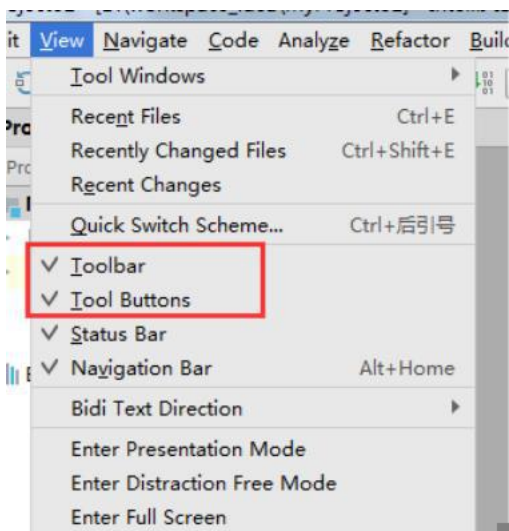
## 2.工程界面展示



- 工程下的 `src` 类似于 Eclipse 下的 `src` 目录。
- 工程下的 `.idea` 和 `Project01.iml` 文件都是 IDEA 工程特有的。类似于 Eclipse 工程下的 `.settings`、`.classpath`、`.project` 等。

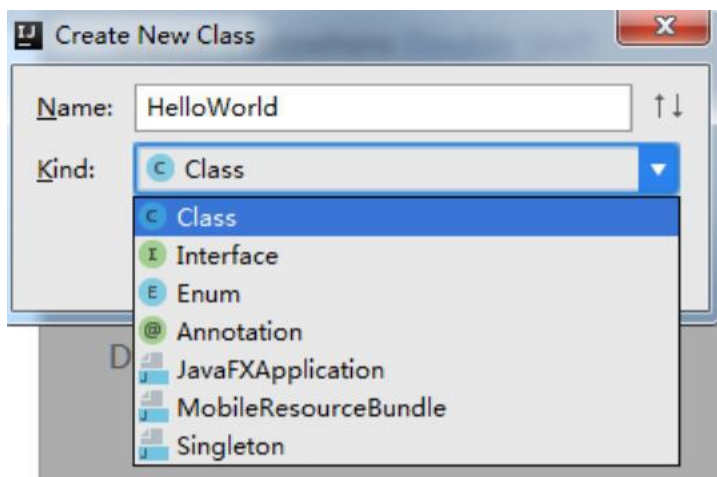
## 3.设置显示常见的视图

(调出面板和按钮组)



## 4.创建 package和 class

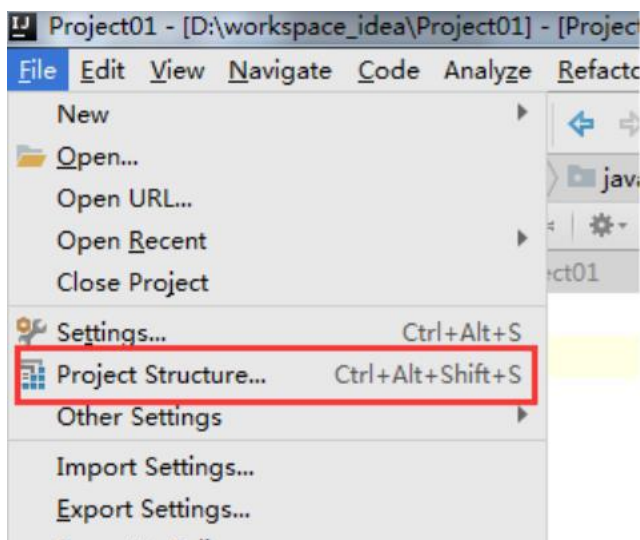
接着在 src目录下创建一个 package，在包下 new-class:



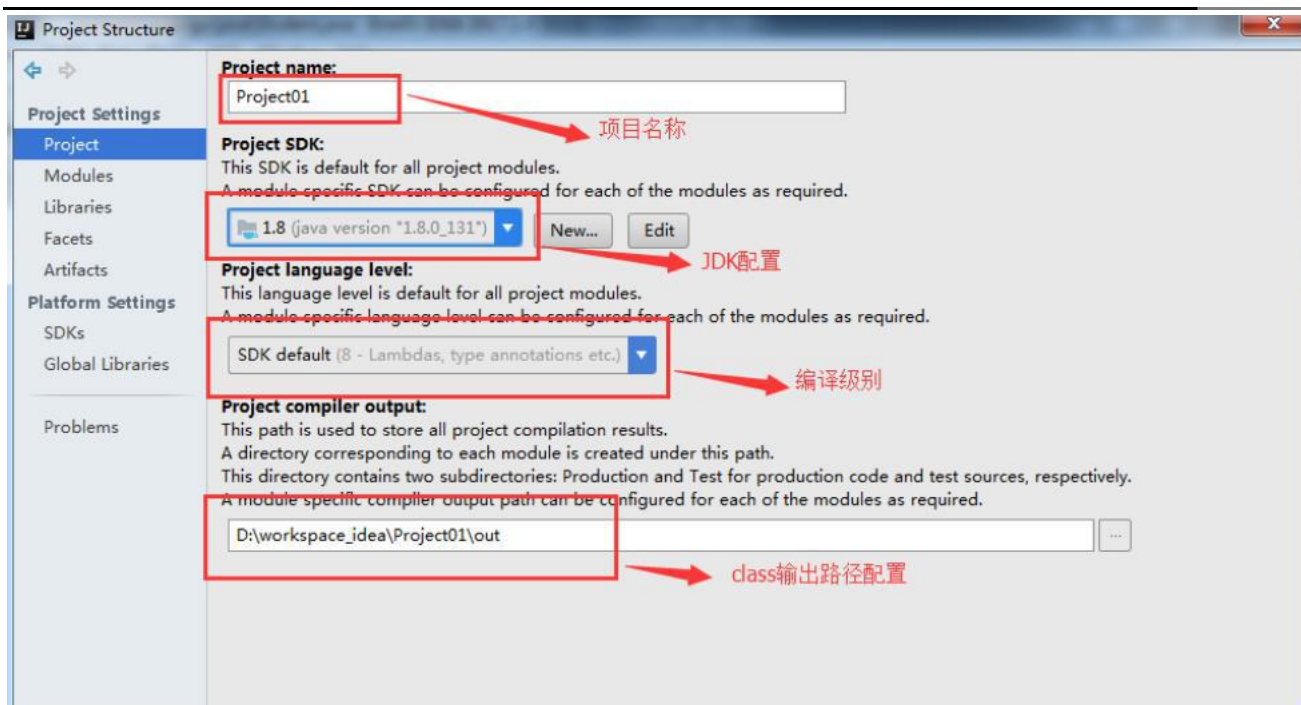
不管是创建 class，还是 interface，还是 annotation，都是选择 new – java class。接着可以在类里声明主方法，输出 helloworld。

说明：在 IDEA里要说的是，写完代码，不用点击保存。IDEA会自动保存代码。

## 5.查看项目配置



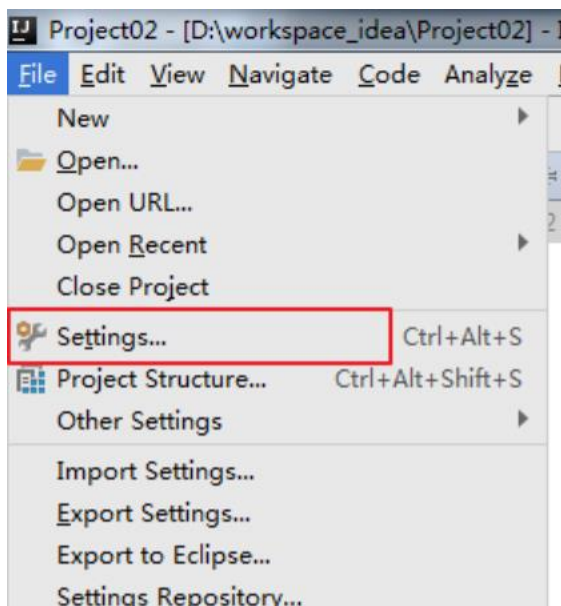
进入项目结构：



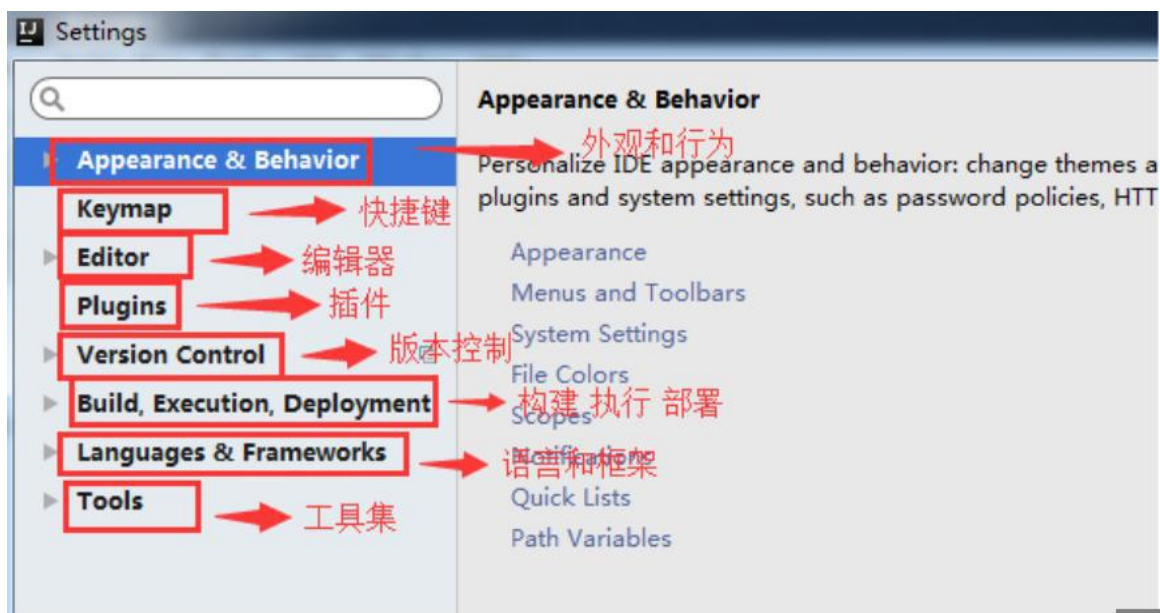
## 五、常用配置

IntelliJ IDEA 有很多人性化的设置我们必须单独拿出来讲解，也因为这些人性化的设置让我们这些 IntelliJ IDEA 死忠粉更加死心塌地使用它和分享它。

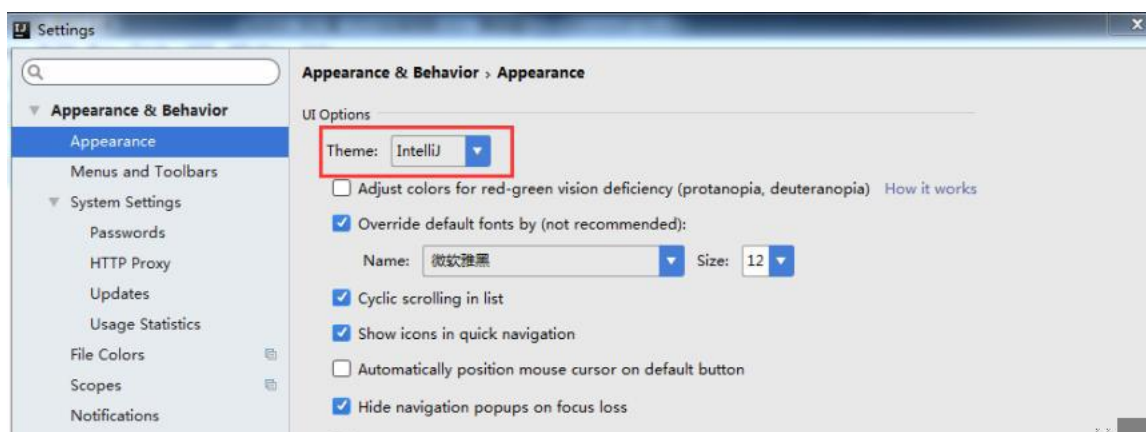
进入设置界面：



目录结构如下：



## 1. 设置主题

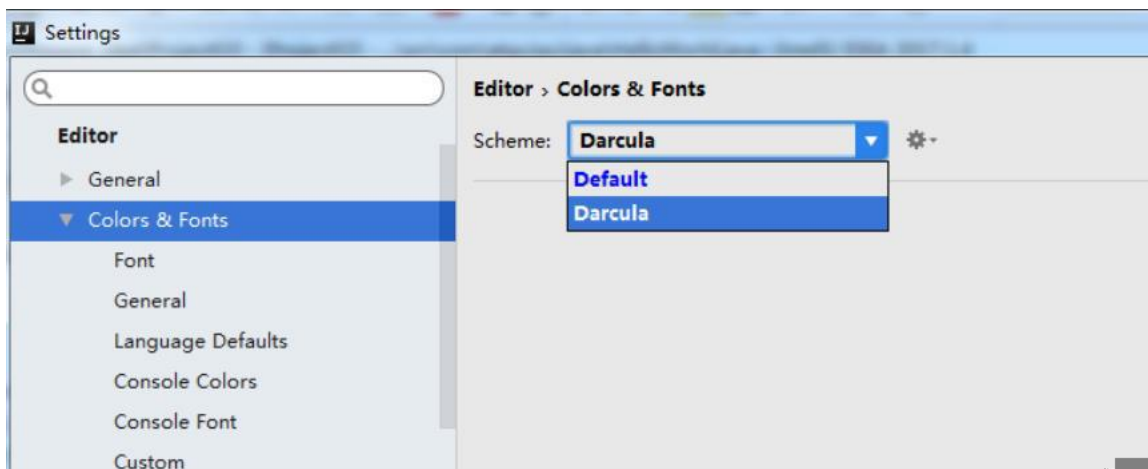


这里默认提供了三套主题：IntelliJ，Darcula，Windows。除了 Darcula 之外，其他都是白色为背景的。改变主题以后，需要重启 IDEA。

## 2. 设置编辑区主题

IDEA 默认提供了两个编辑区主题，可以通过如下的方式进行选择。





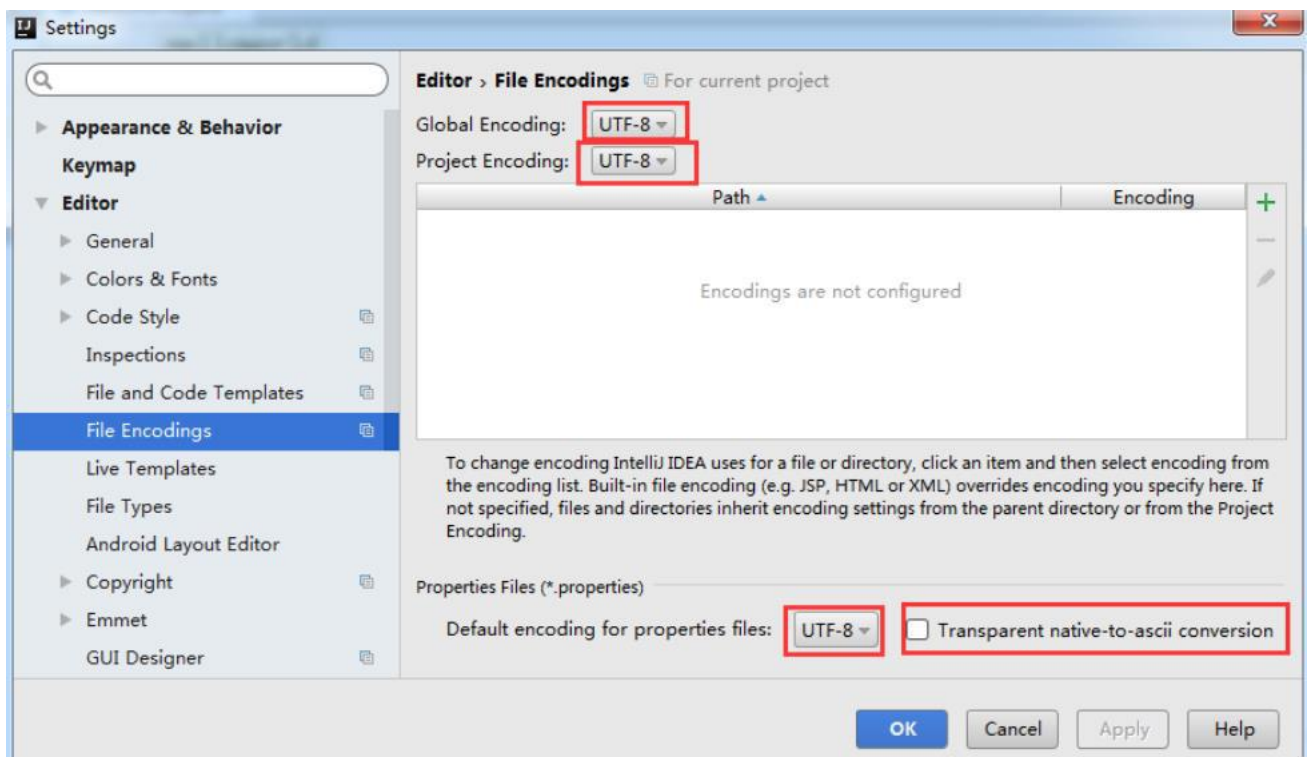
如果想要更多的主题效果的话，可以到如下的网站下载：

<http://www.riaway.com/>

下载以后，导入主题：

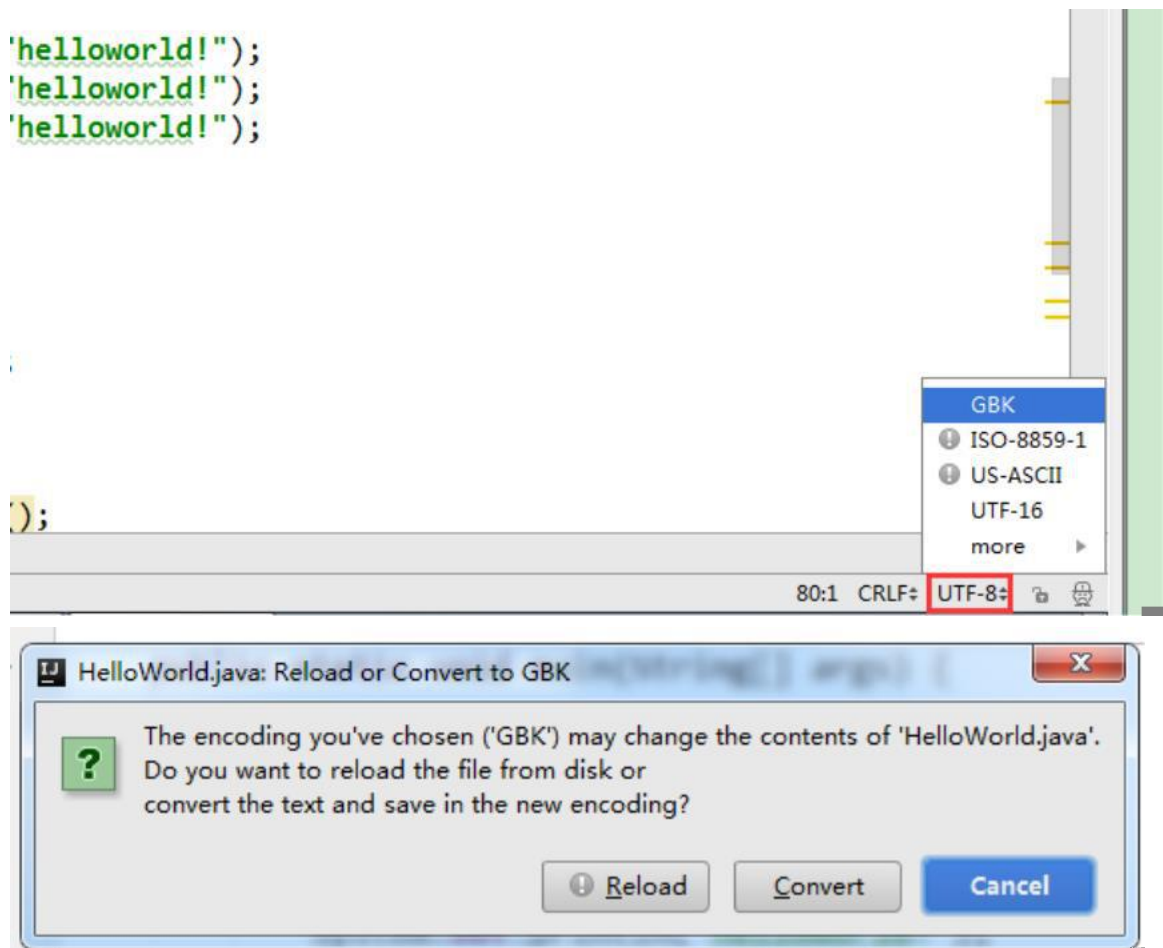
file -> import settings -> 选中下载的主题 jar 文件 -> 一路确认 -> 重启。

### 3. 设置项目文件编码



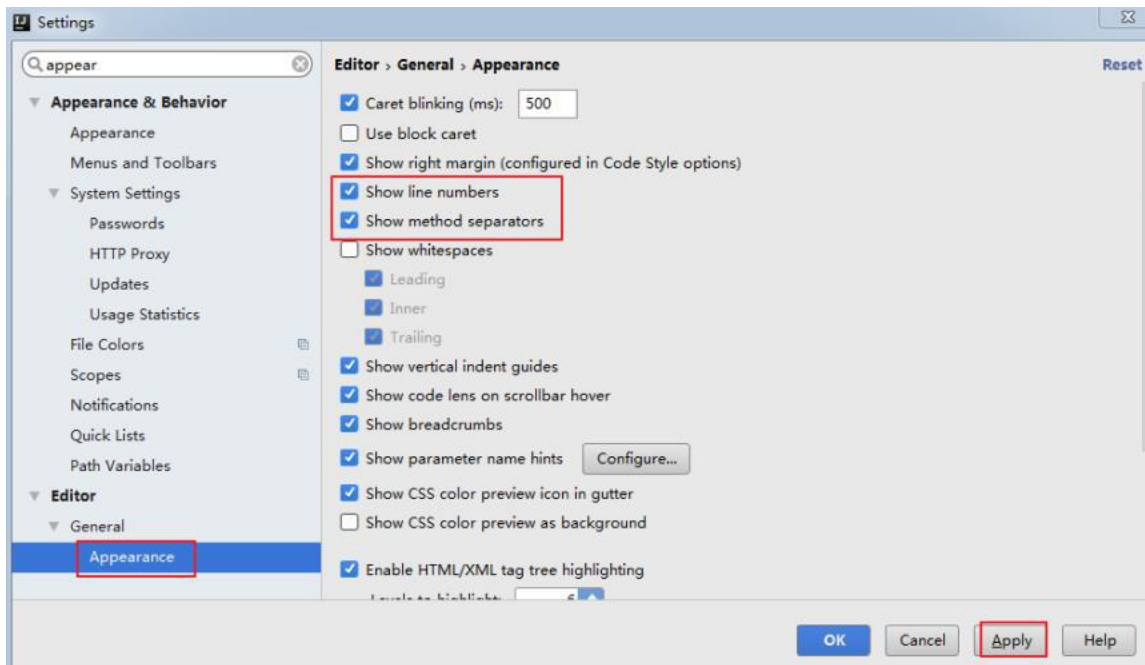
说明：Transparent native-to-ascii conversion 主要用于转换 `ascii`，一般都要勾选，不然 Properties 文件中的注释显示的都不会是中文。

#### 4. 设置当前源文件的编码



对单独文件的编码修改还可以点击右下角的编码设置区。如果代码内容中包含中文，则会弹出如上的操作选择。其中，①**Reload**表示使用新编码重新加载，新编码不会保存到文件中，重新打开此文件，旧编码是什么依旧还是什么。②**Convert**表示使用新编码进行转换，新编码会保存到文件中，重新打开此文件，新编码是什么则是什么。③含有中文的代码文件，**Convert**之后可能会使中文变成乱码，所以在转换成请做好备份，不然可能出现转换过程变成乱码，无法还原。

## 5. 设置显示行号和方法间的分隔符



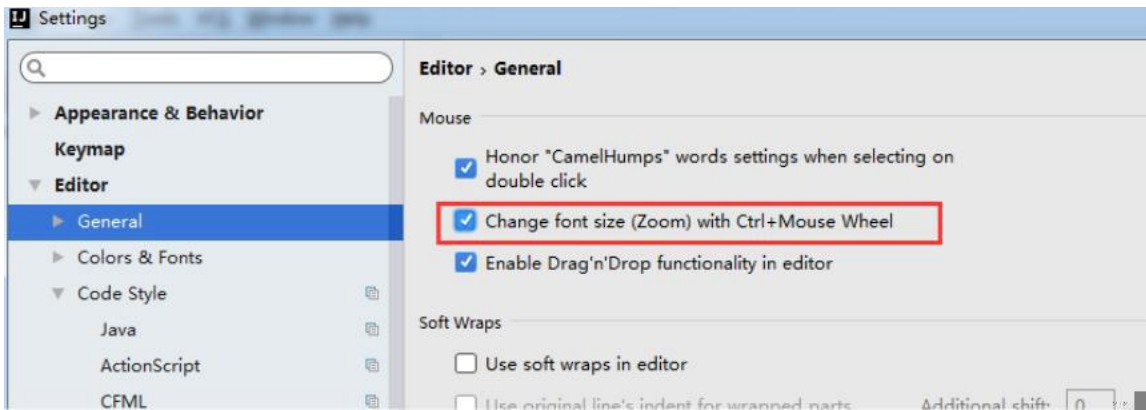
如上图红圈所示，默认 IntelliJ IDEA 是没有勾选 `Show line numbers` 显示行数的，但是我建议一般这个要勾选上。

如上图红圈所示，默认 IntelliJ IDEA 是没有勾选 `Show method separators` 显示方法线的，这种线有助于我们区分开方法，所以也是建议勾选上的。

## 6. 设置字体、字体大小、字体行间距

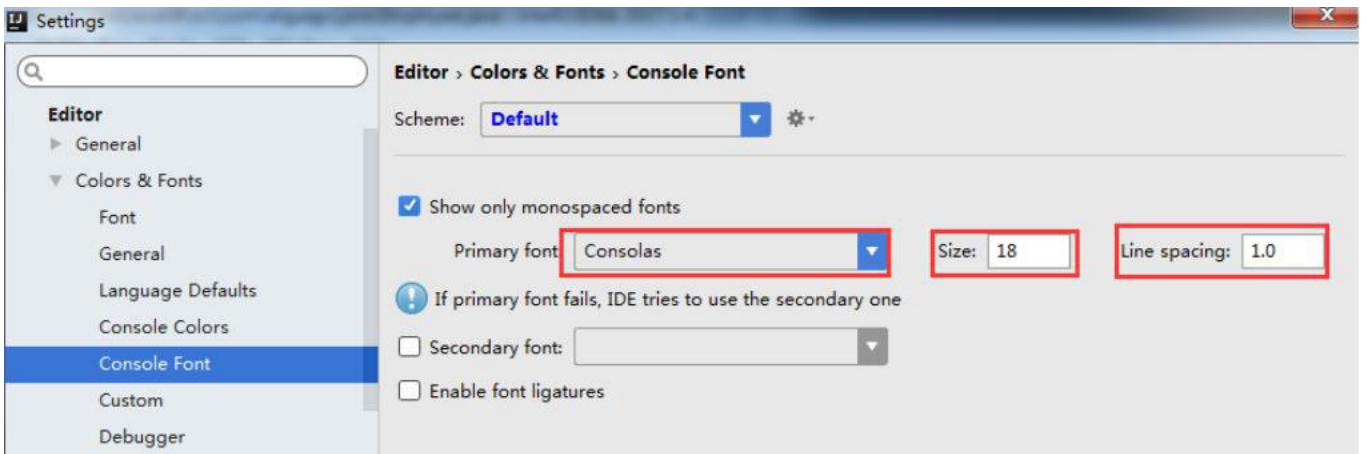


## 7. 设置鼠标滚轮修改字体大小

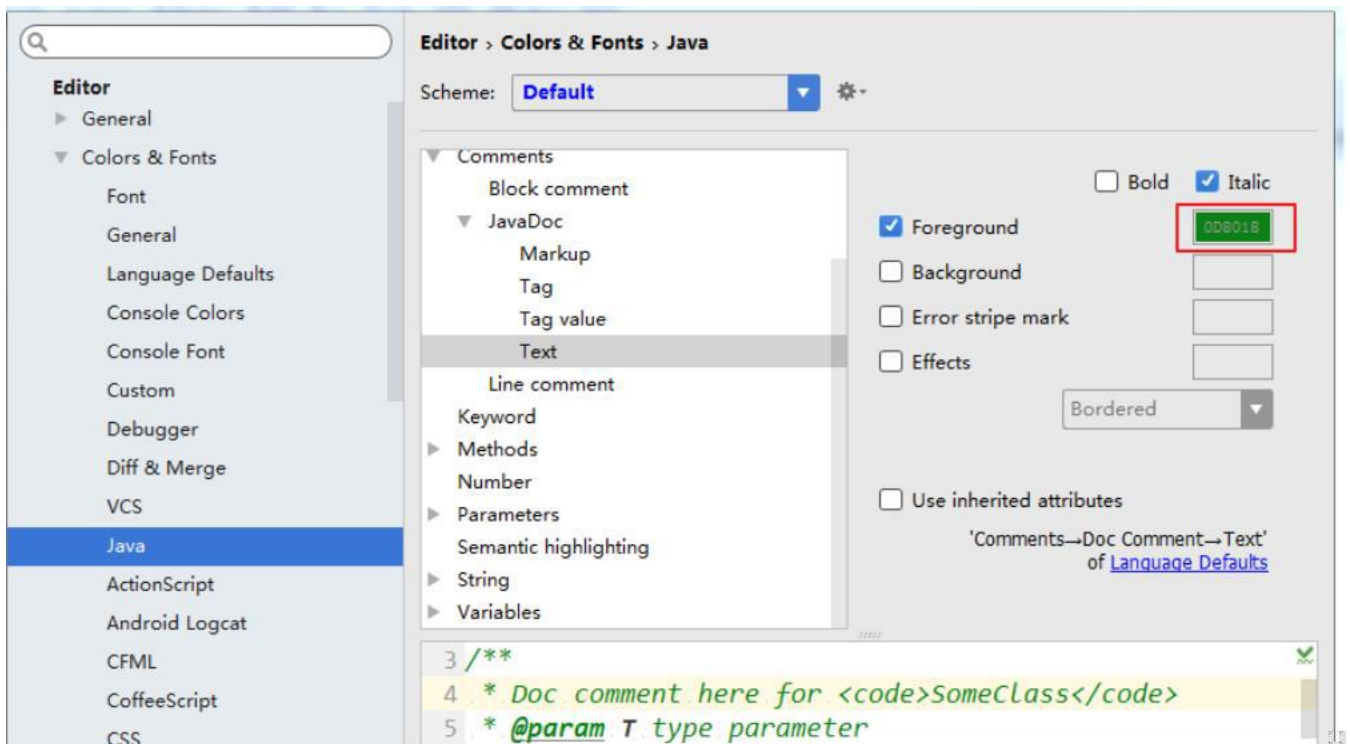


我们可以勾选此设置后，增加 **Ctrl + 鼠标滚轮** 快捷键来控制代码字体大小显示。

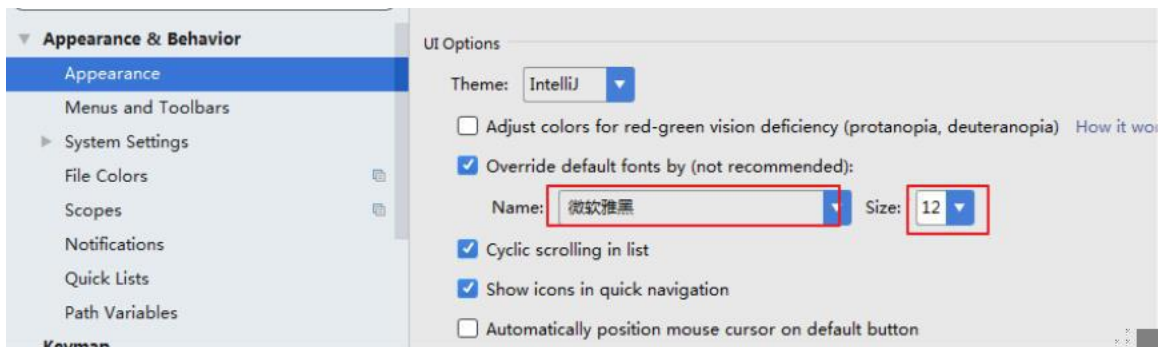
## 8. 修改控制台输出的字体及字体大小



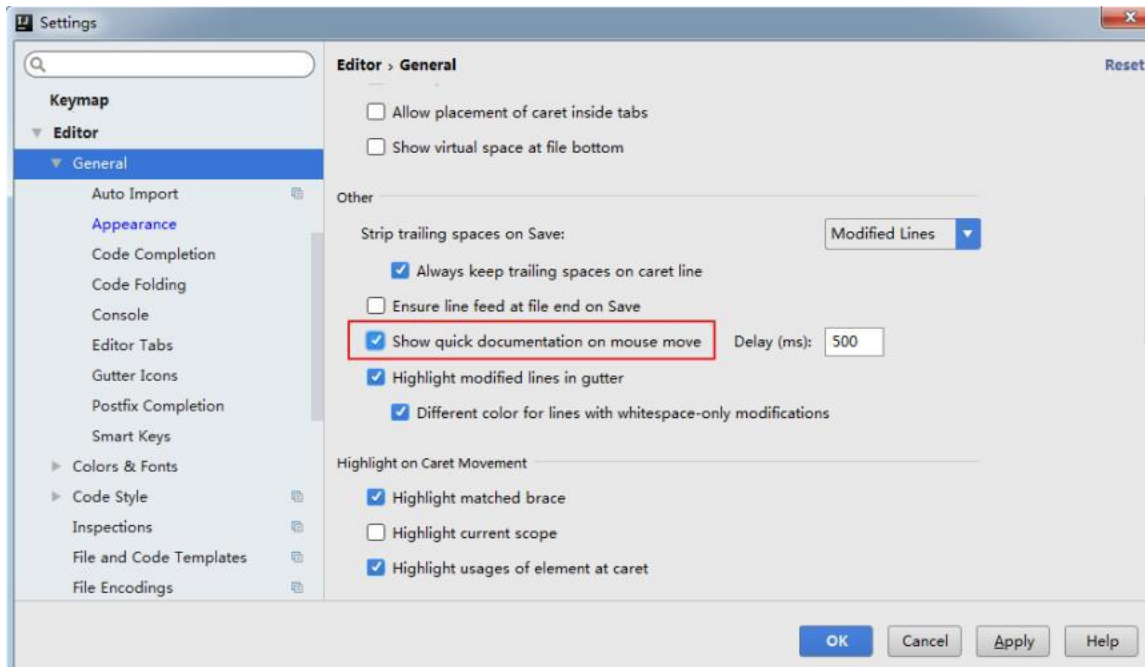
## 9. 修改文档注释的字体颜色



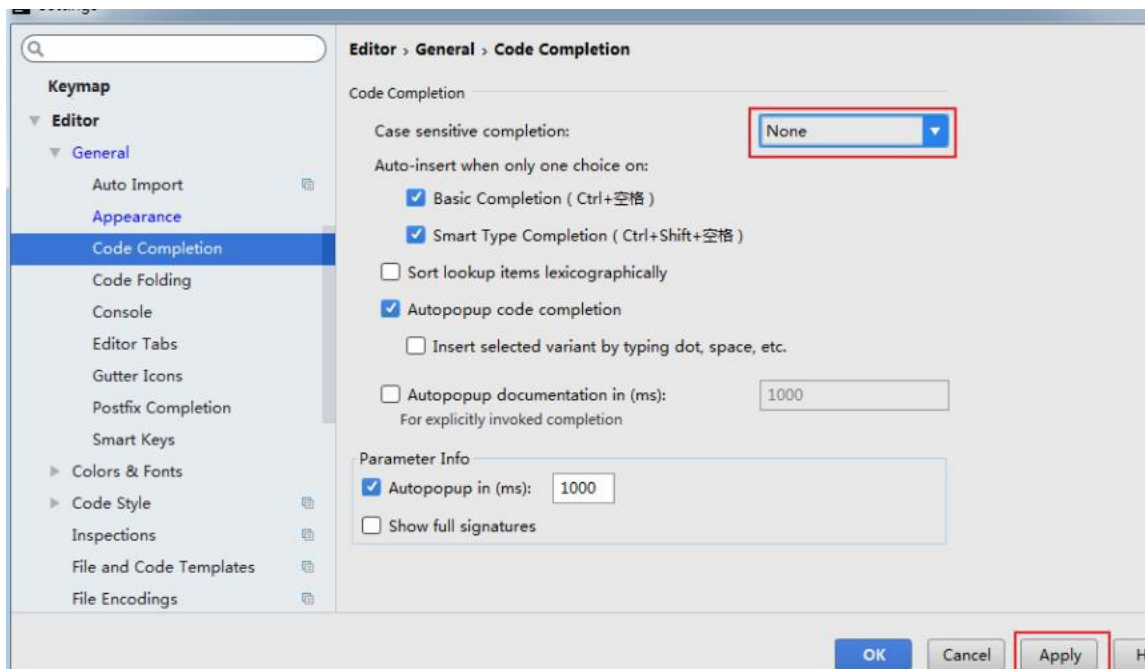
## 10. 设置窗体及菜单的字体及字体大小（可不设置）



## 11. 设置鼠标悬浮提示



## 12. 忽略大小写提示



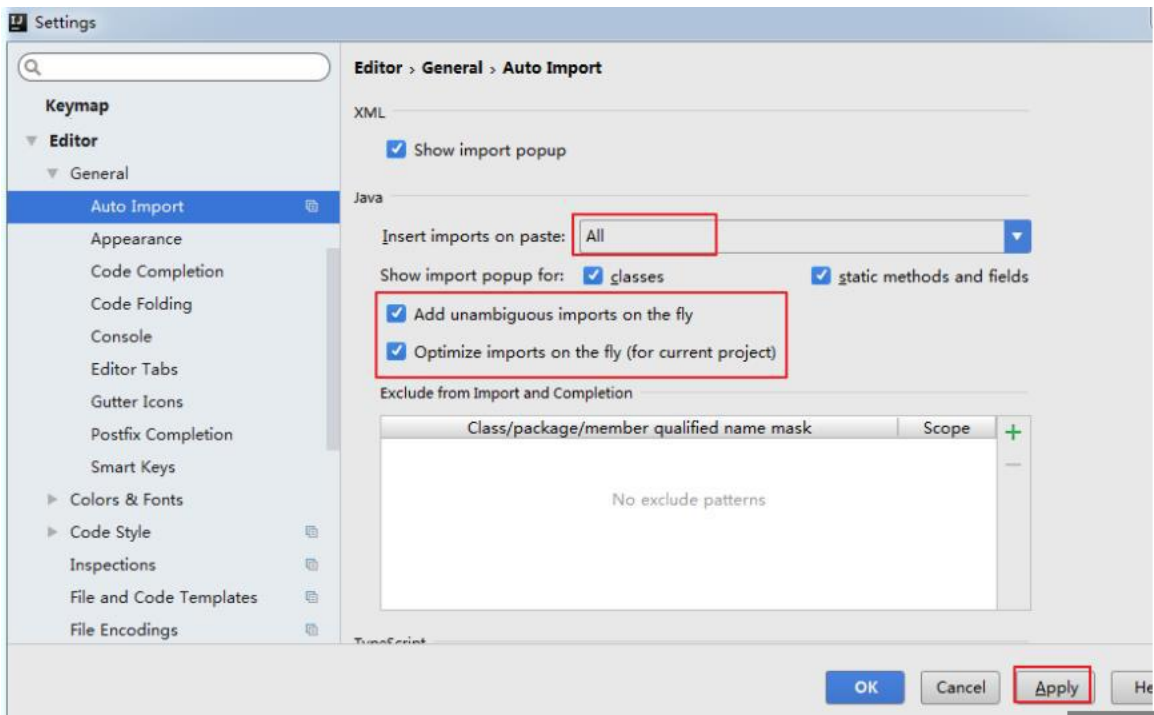
- IntelliJ IDEA的代码提示和补充功能有一个特性：区分大小写。如上图标注所示，默认就是 First letter区分大小写的。
- 区分大小写的情况是这样的：比如我们在 Java代码文件中输入 stringBuffer，IntelliJ IDEA是不会帮



我们提示或是代码补充的，但是如果输入 `StringBuffer` 就可以进行代码提示和补充。

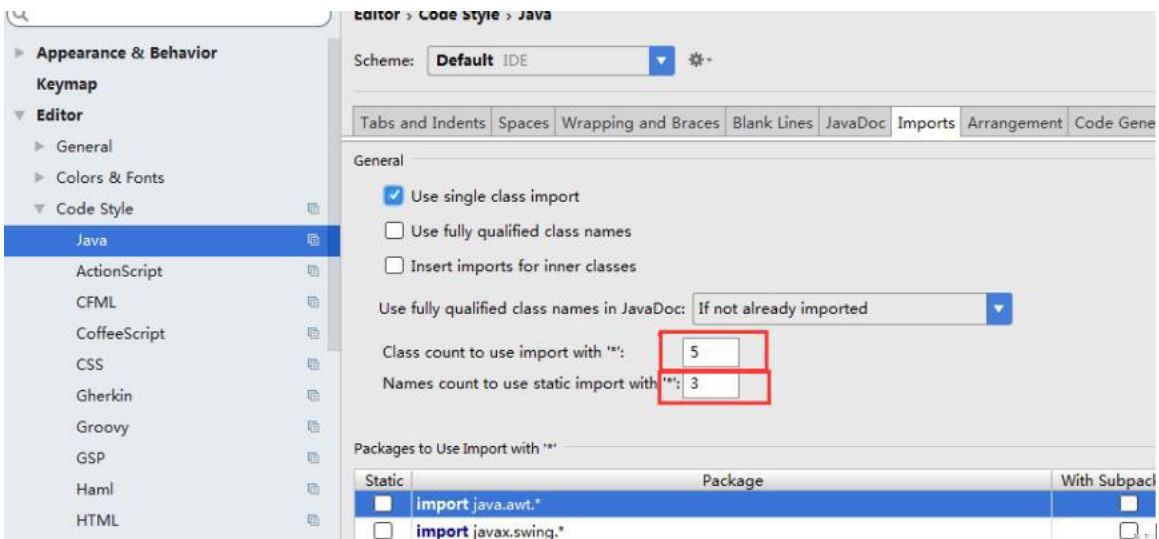
- 如果想不区分大小写的话，改为 `None` 选项即可。

## 13. 设置自动导包功能



- Add unambiguous imports on the fly: 自动帮我们优化导入的包
- Optimize imports on the fly: 自动去掉一些没有用到的包

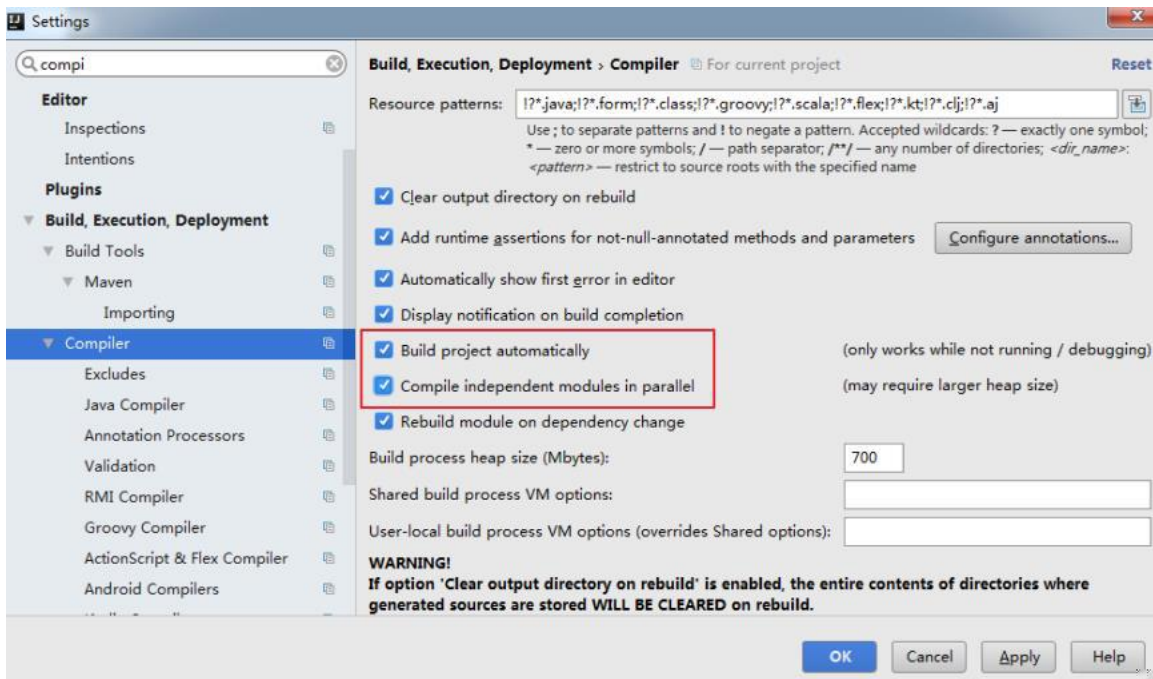
## 14. 设置超过指定 import 个数，改为\*



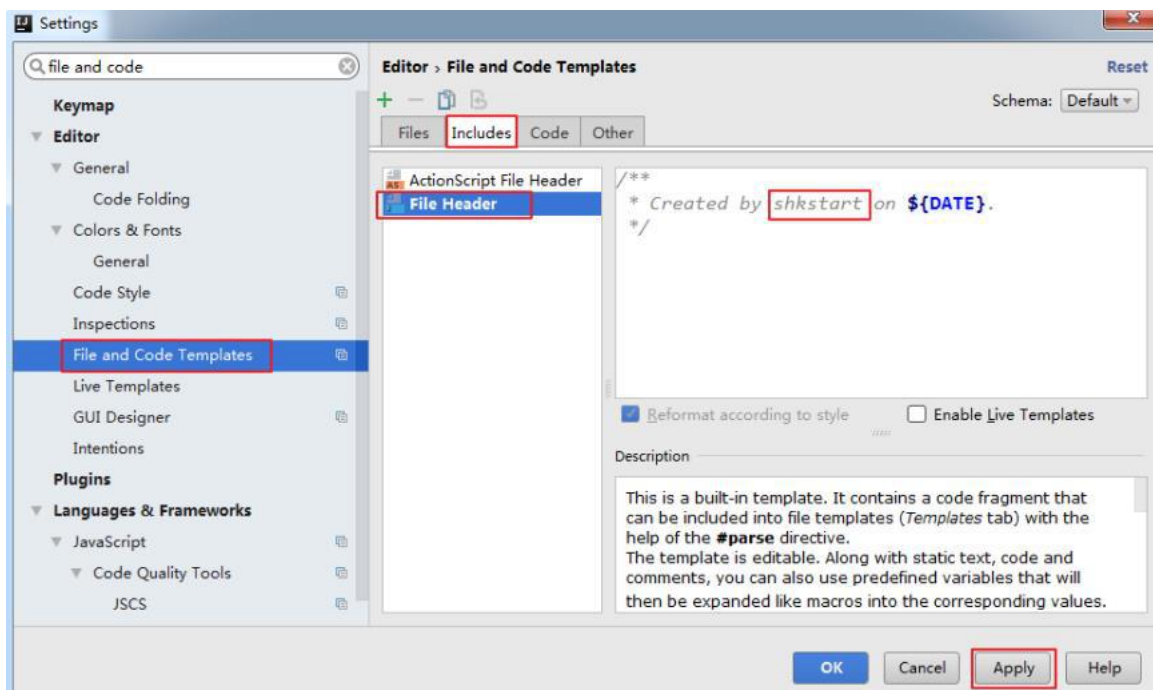


## 15. 设置自动编译

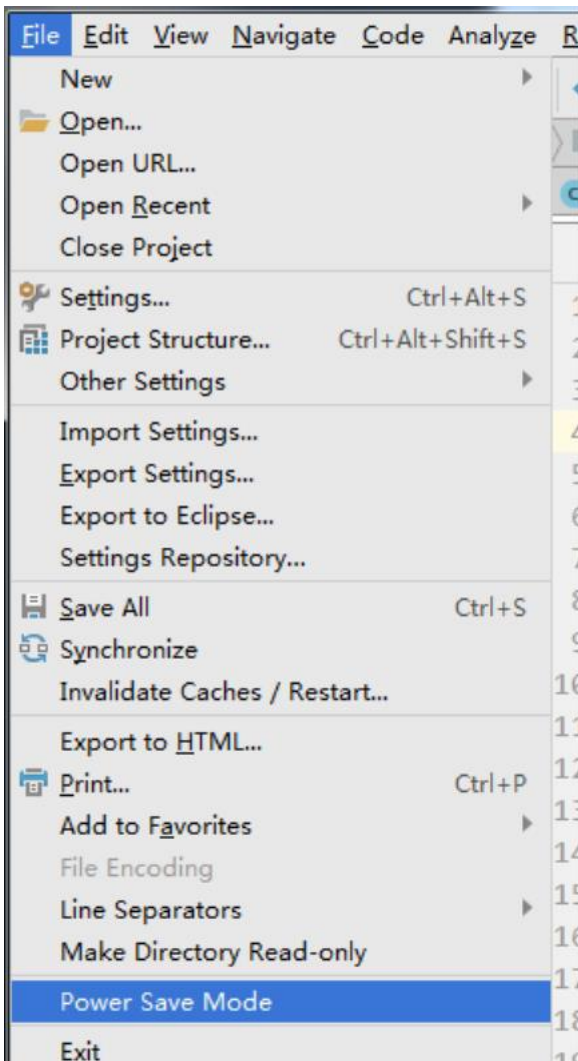
IntelliJ Idea默认状态为不自动编译状态，Eclipse默认为自动编译，很多朋友都是从 Eclipse转到 IntelliJ的，这常常导致我们在需要操作 class文件时忘记对 java类文件进行编译从而对旧文件进行了操作。



## 16. 修改类头的文档注释信息

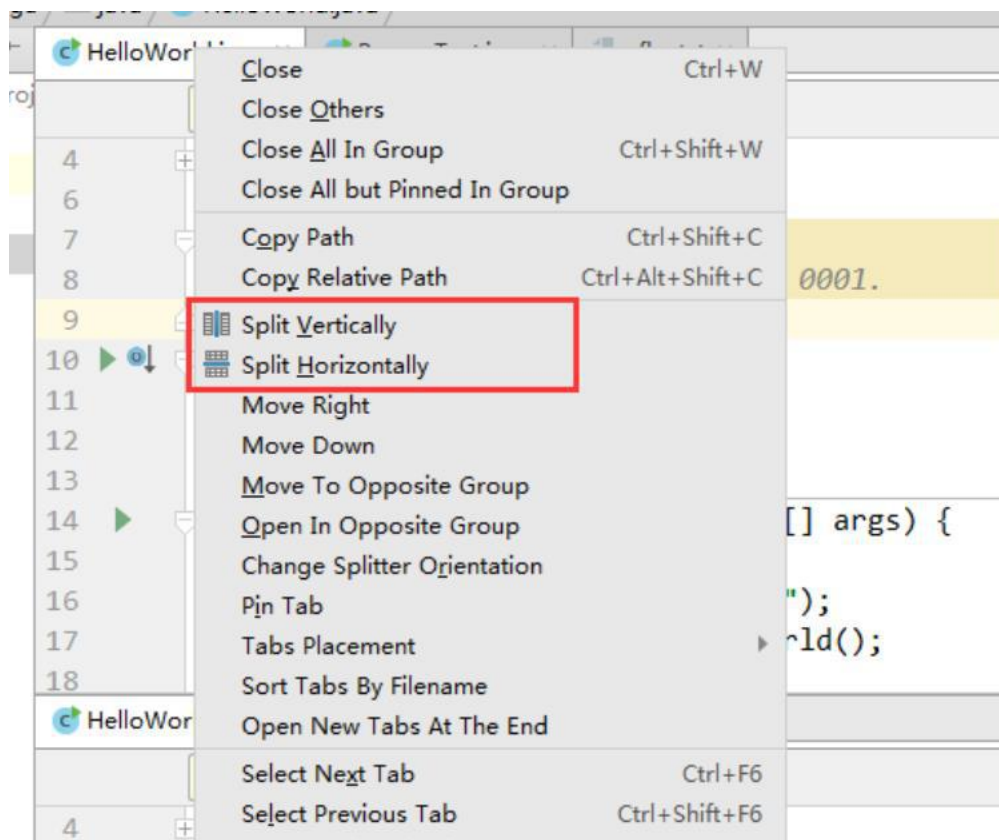


## 17. 设置为省电模式

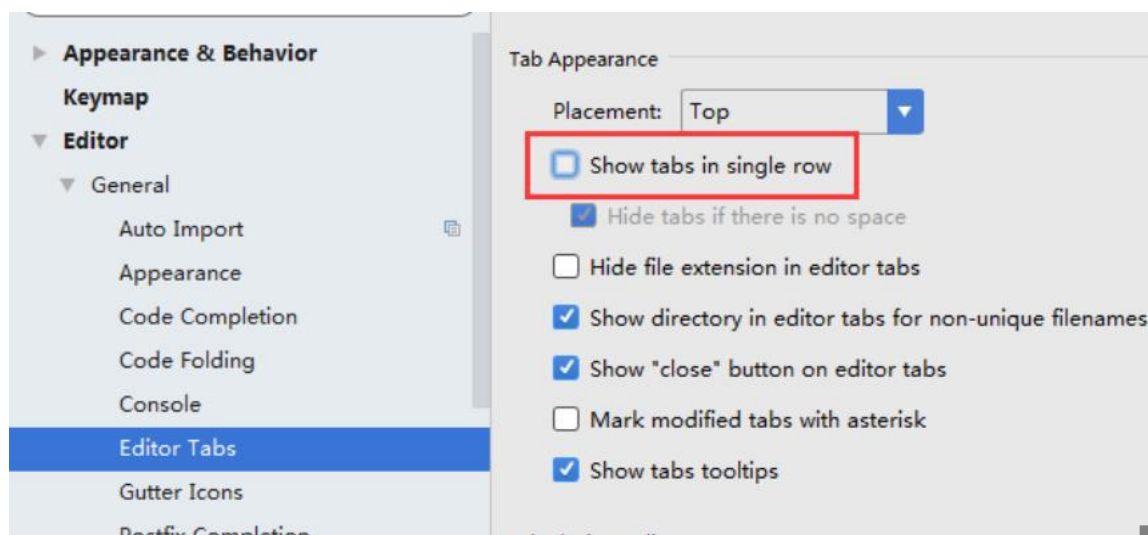


如上图所示，IntelliJ IDEA有一种叫做**省电模式**的状态，开启这种模式之后 IntelliJ IDEA会关掉代码检查和代码提示等功能。所以一般我也会认为这是一种**阅读模式**，如果你在开发过程中遇到突然代码文件不能进行检查和提示可以来看看这里是否有开启该功能。

## 18. 设置代码水平或垂直显示



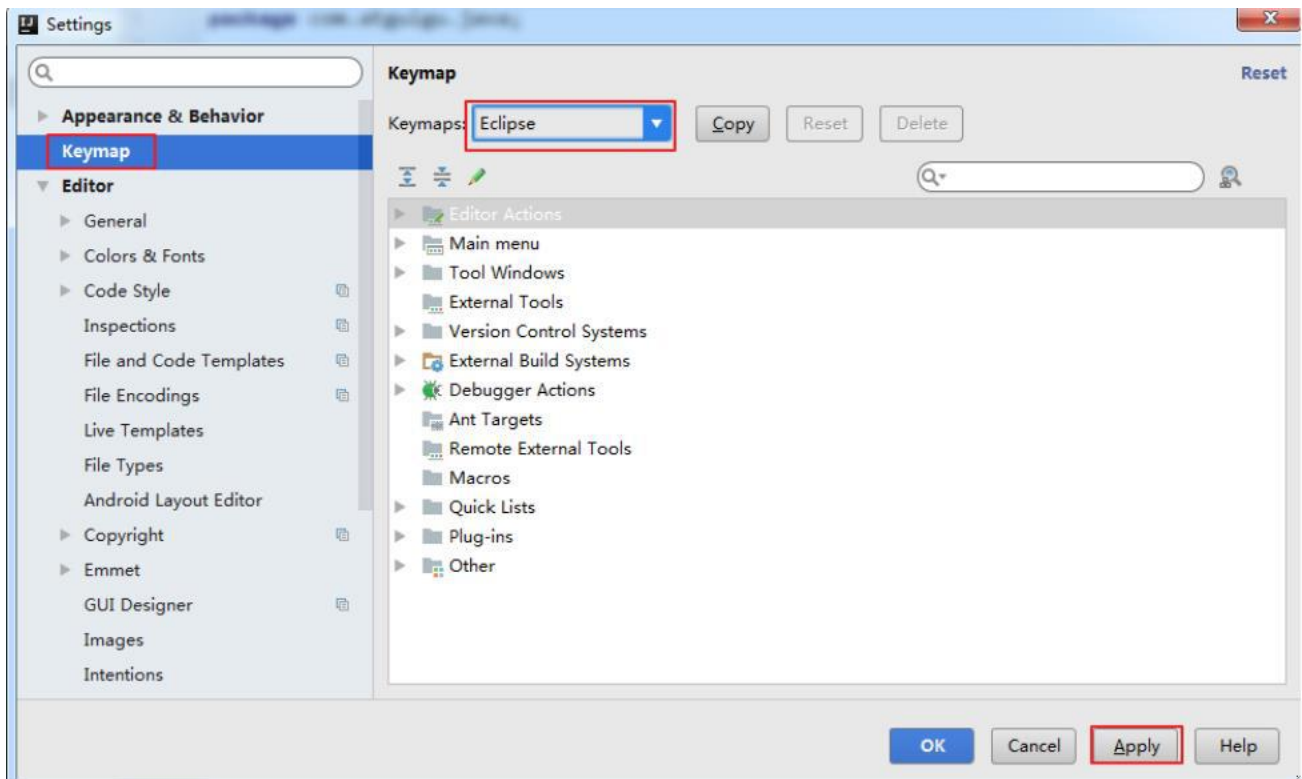
## 19. 设置取消单行显示 tabs 的操作



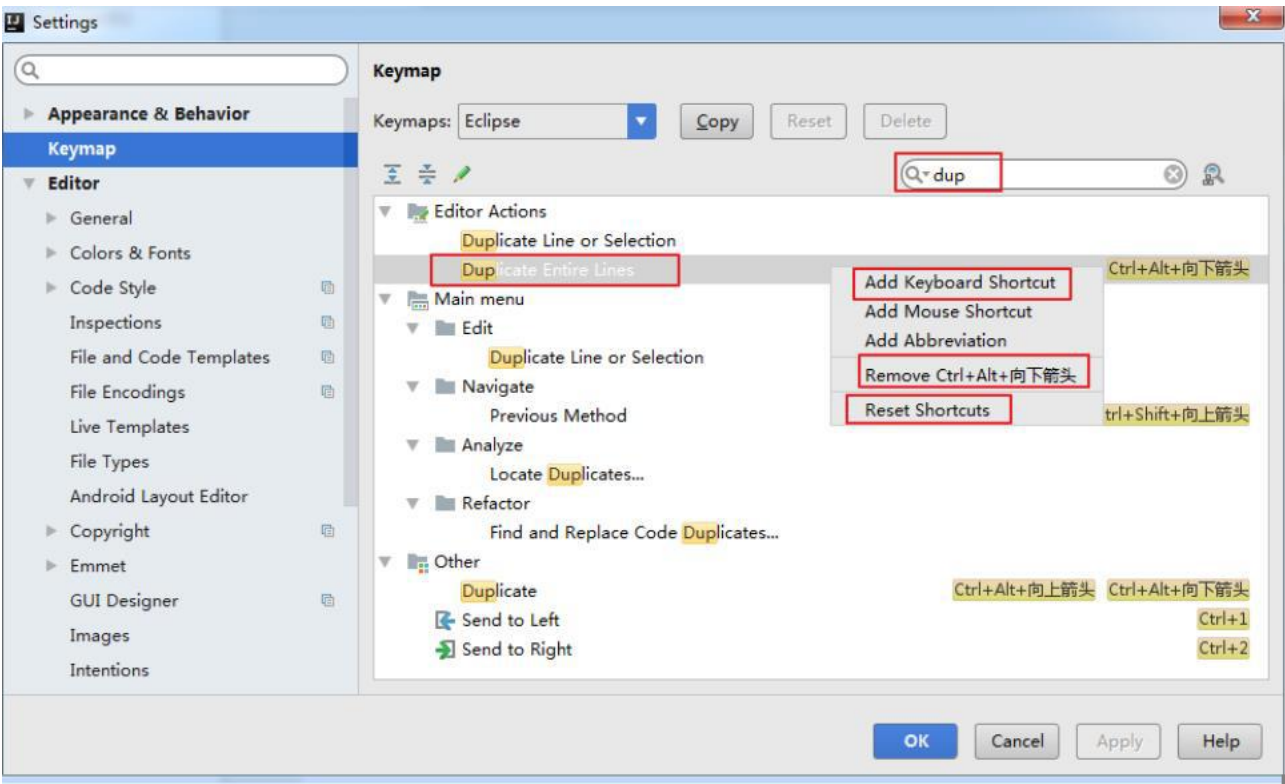
如上图标注所示，在打开很多文件的时候，IntelliJ IDEA默认是把所有打开的文件名 Tab单行显示的。但是我个人现在的习惯是使用多行，多行效率比单行高，因为单行会隐藏超过界面部分 Tab，这样找文件不方便。

## 六、设置快捷键

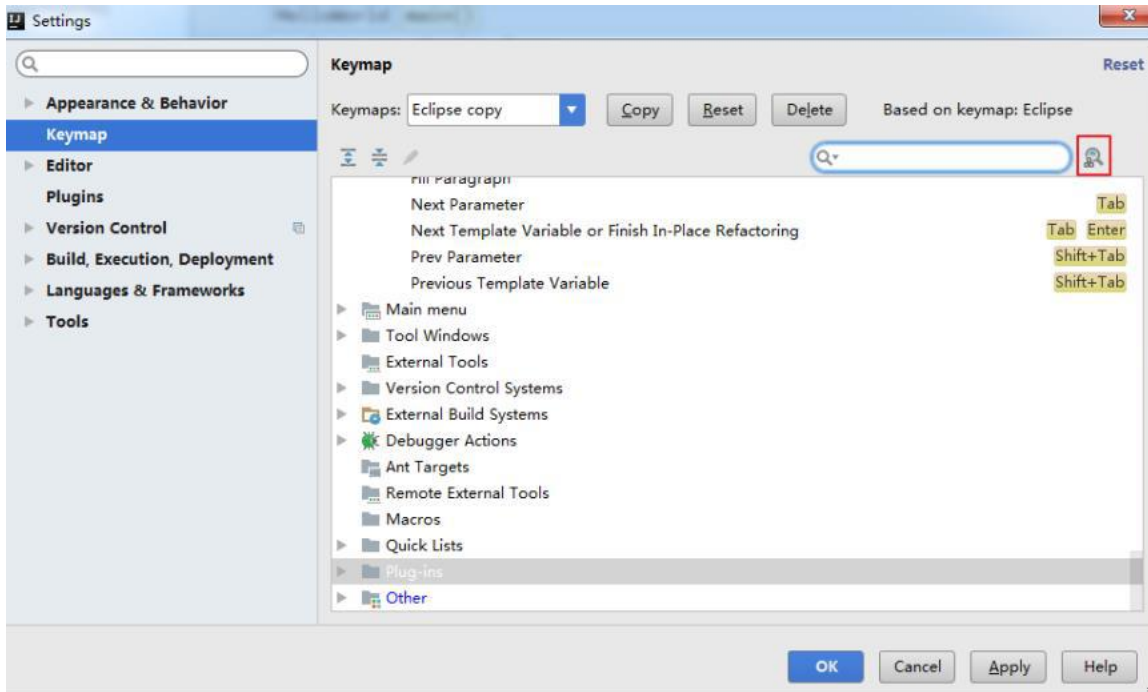
### 1. 设置快捷为 Eclipse 的快捷键



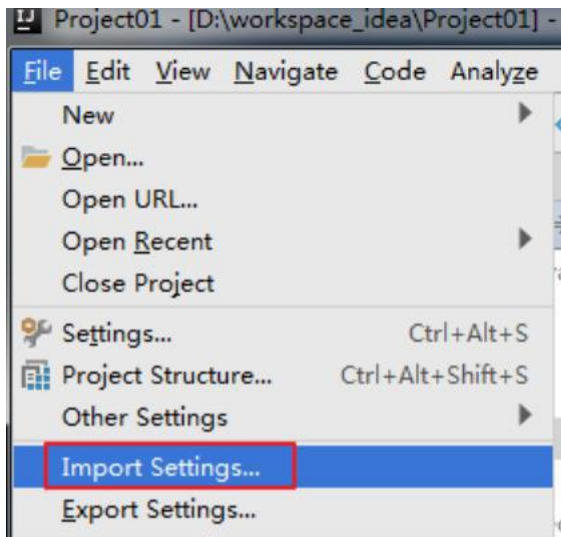
## 2.修改快捷键



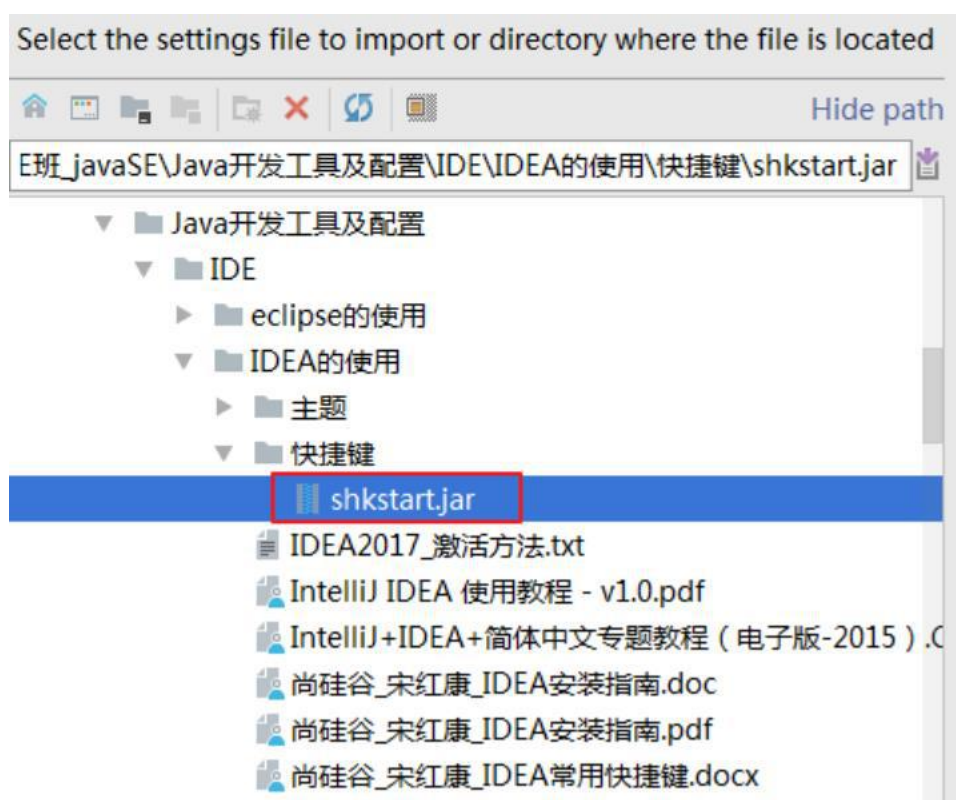
### 3. 查找指定快捷键的作用位置



### 4. 导入已有的设置







点击 OK 之后，重启 IDEA 即可。

## 5. 常用快捷键

—————尚硅谷·宋红康设置版

- |  |                  |                    |
|--|------------------|--------------------|
| 1. 执行(run)                                   | alt+r            |                    |
| 2. 提示补全 (Class Name Completion)              | alt+/            |                    |
| 3. 单选注释                                      | ctrl + /         |                    |
| 4. 多行注释                                      | ctrl + shift + / |                    |
| 5. 向下复制一行 (Duplicate Lines)                  | ctrl+alt+down    |                    |
| 6. 删除一行或选中行 (delete line)                    | ctrl+d           |                    |
| 7. 向下移动行(move statement down)                | alt+down         |                    |
| 8. 向上移动行(move statement up)                  | alt+up           |                    |
| 9. 向下开始新的一行(start new line)                  | shift+enter      |                    |
| 10. 向上开始新的一行 (Start New Line before current) | ctrl+shift+enter |                    |
| 11. 如何查看源码 (class)                           | ctrl + 选中指定的结构   | 或 ctrl + shift + t |
| 12. 万能解错/生成返回值变量                             | alt + enter      |                    |



---

13.退回到前一个编辑的页面 (back)	alt + left
14.进入到下一个编辑的页面(针对于上面那条来说的) (forward)	alt + right
15.查看类继承关系(type hierarchy)	F4
16.格式化代码(reformat code)	ctrl+shift+F
17.提示方法参数类型(Parameter Info)	ctrl+alt+/
18.复制代码	ctrl + c
19.撤销	ctrl + z
20.反撤销	ctrl + y
21.剪切	ctrl + x
22.粘贴	ctrl + v
23.保存	ctrl + s
24.全选	ctrl + a
25.选中数行，整体往后移动	tab
26.选中数行，整体往前移动	shift + tab
27.查看类的结构：类似于 eclipse的 outline	ctrl+o
28.修改变量名与方法名(rename)	alt+shift+r
29.大写转小写/小写转大写(toggle case)	ctrl+shift+y
30.生成构造/get/set/toString	alt +shift + s
31.查看文档说明(quick documentation)	F2
32.收起所有的方法(collapse all)	alt + shift + c
33.打开所有方法(expand all)	alt+shift+x
34.打开代码所在硬盘文件夹(show in explorer)	ctrl+shift+x
35.生成 try-catch等(surround with)	alt+shift+z
36.局部变量抽取为成员变量(introduce field)	alt+shift+F
37.查找/替换(当前)	ctrl+f
38.查找(全局)	ctrl+h
39.查找文件	double Shift

---

40.抽取方法(Extract Method)	alt+shift+m
41.打开 Module setting窗口(Recently Changed Files)	ctrl+E
42.关闭当前打开的代码栏(close)	ctrl + w
43.关闭打开的所有代码栏(close all)	ctrl + shift + w
44.快速搜索类中的错误(next highlighted error)	ctrl + shift + Q
45.选择要粘贴的内容(Show in Explorer)	ctrl+shift+V
46.查找方法在哪里被调用(Call Hierarchy)	ctrl+shift+H
47.查看方法的多层重写结构(method hierarchy)	ctrl+alt+h

## 七、关于模板

### 1. sout : System.out.println()快捷输出

类似的:

```
soutp=System.out.println("方法形参名 =" + 形参名);
```

```
soutv=System.out.println("变量名 =" + 变量);
```

```
soutm=System.out.println("当前类名.当前方法");
```

```
"abc".sout => System.out.println("abc");
```

### 2.list.for :可生成集合 list的 for循环

```
List<String> list = new ArrayList<String>();
```

输入: list.for 即可输出

```
for(String s:list){
```

```
}
```

---

### 3. `psvm` :可生成 `main`方法

### 4. `fori` :可生成 `for`循环

类似的:

`iter`:生成增强 `for`循环

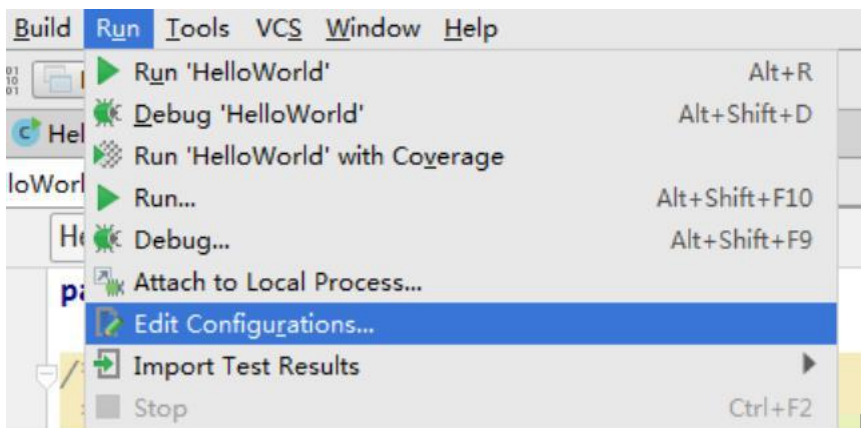
`itar`:生成普通 `for`循环

具体如何设置 **IDEA**的 **Live Templates**, 可参见《尚硅谷\_宋红康\_IDEA的代码模板.pdf》

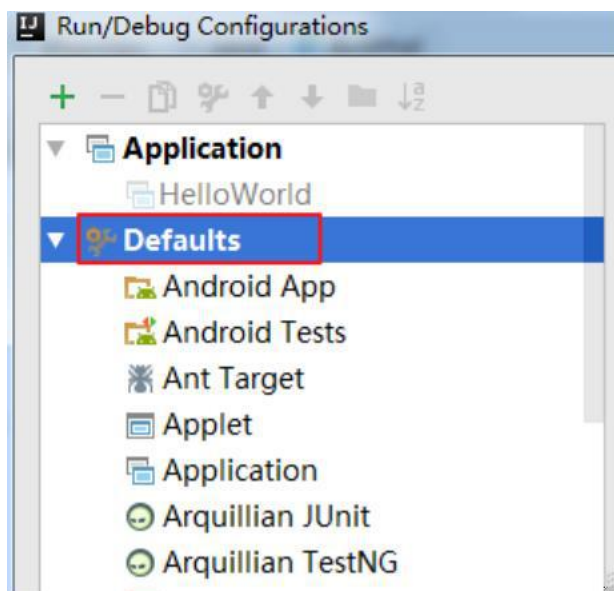
## 八、配置 **Tomcat**

在 **IDEA**中配置 **Tomcat**之前, 需要保证已经安装并配置了 **Tomcat**的环境变量。如果没有安装并配置, 可以参考《尚硅谷\_宋红康\_Tomcat快速部署.pdf》,配置完成以后, 在命令行输入:  
**catalina run**。能够启动 **tomcat**, 则证明安装配置成功。

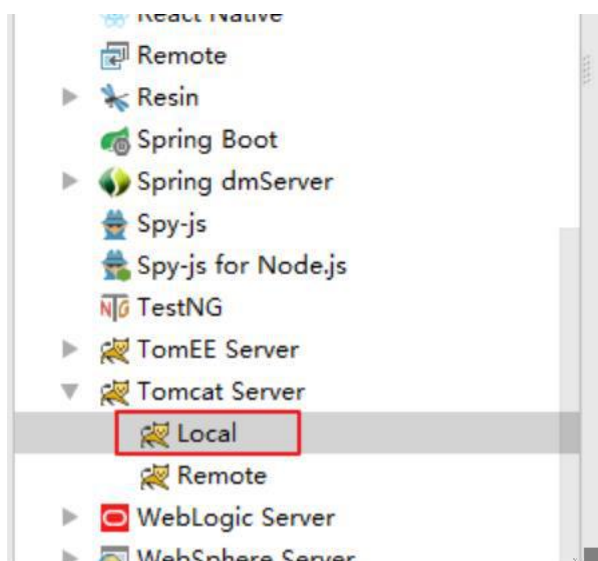
下面看如何在 **IDEA**中配置:



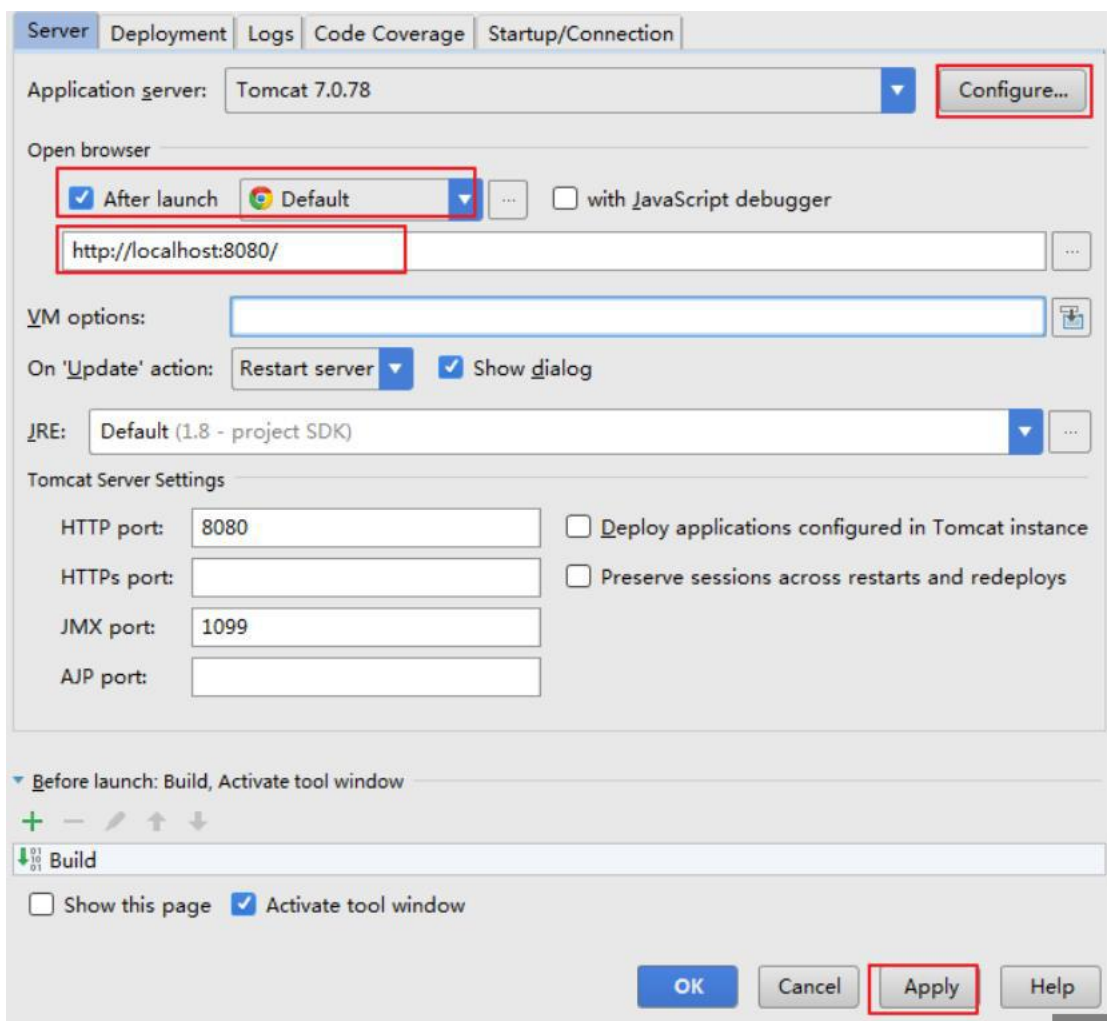
点击 **Edit Configurations**



选择 Defaults，然后下拉进度条：

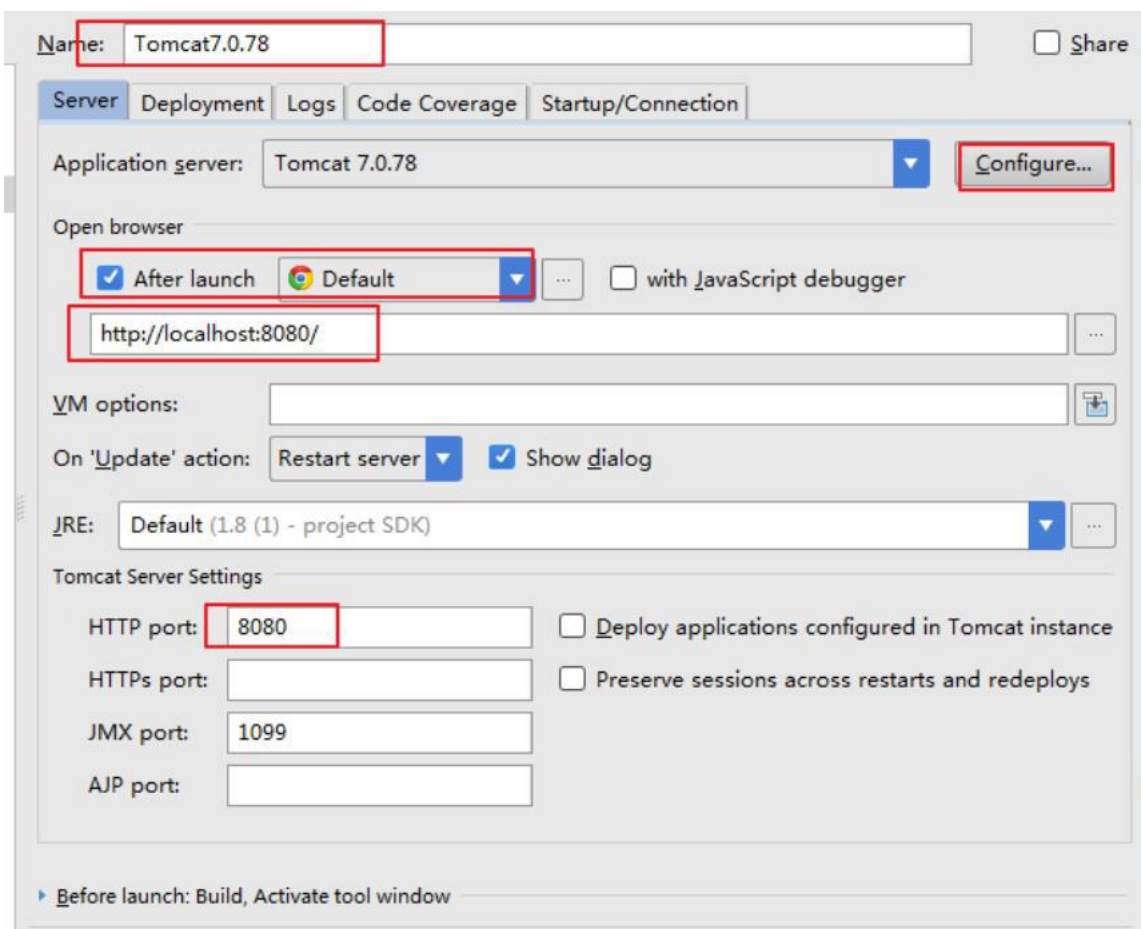
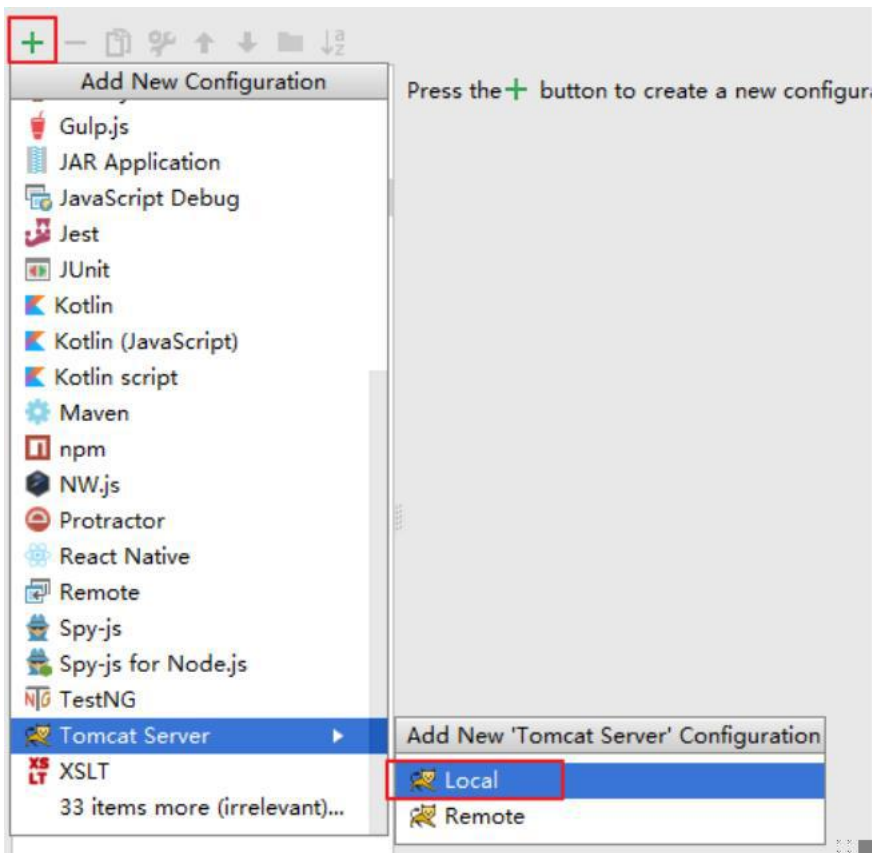


这里选择 TomEE Server或者 Tomcat Server都可以。接着选择 Local。

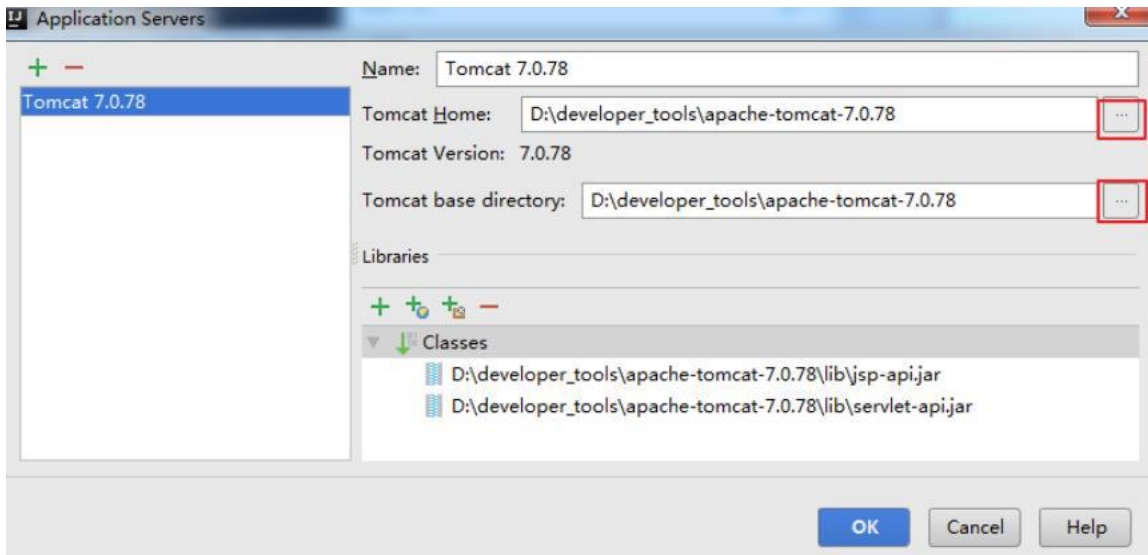


点击 apply保存设置。

继续：



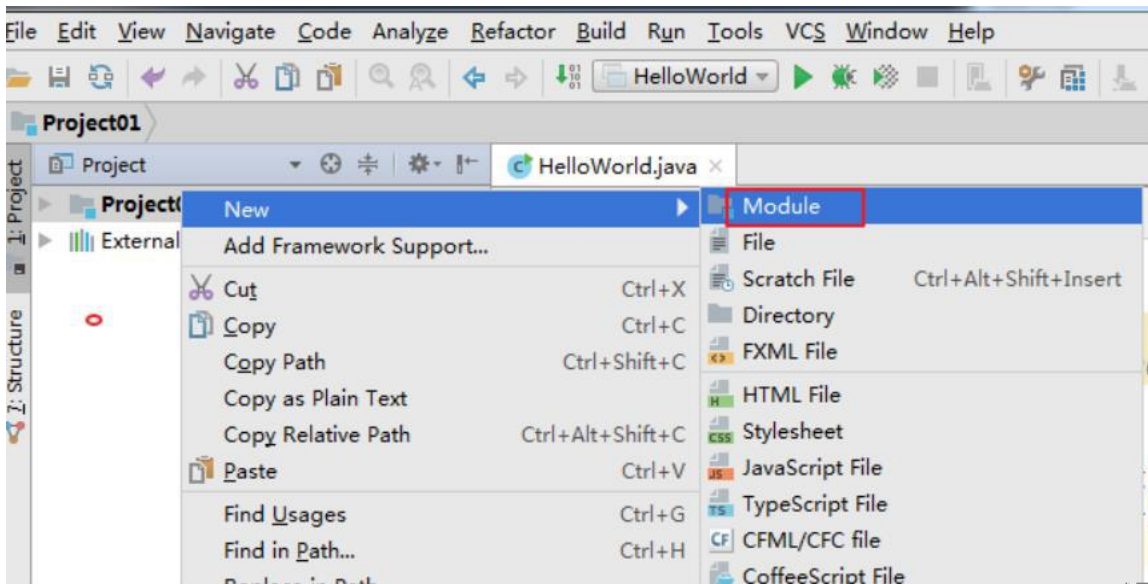
这里给配置 Tomcat命名。设置要启动的浏览器以及端口号等。  
其中，点击 Configure按钮，选择 Tomcat的安装目录：



点击 OK即可。

## 九、创建 Web工程并运行

工程栏空白处 new – module:



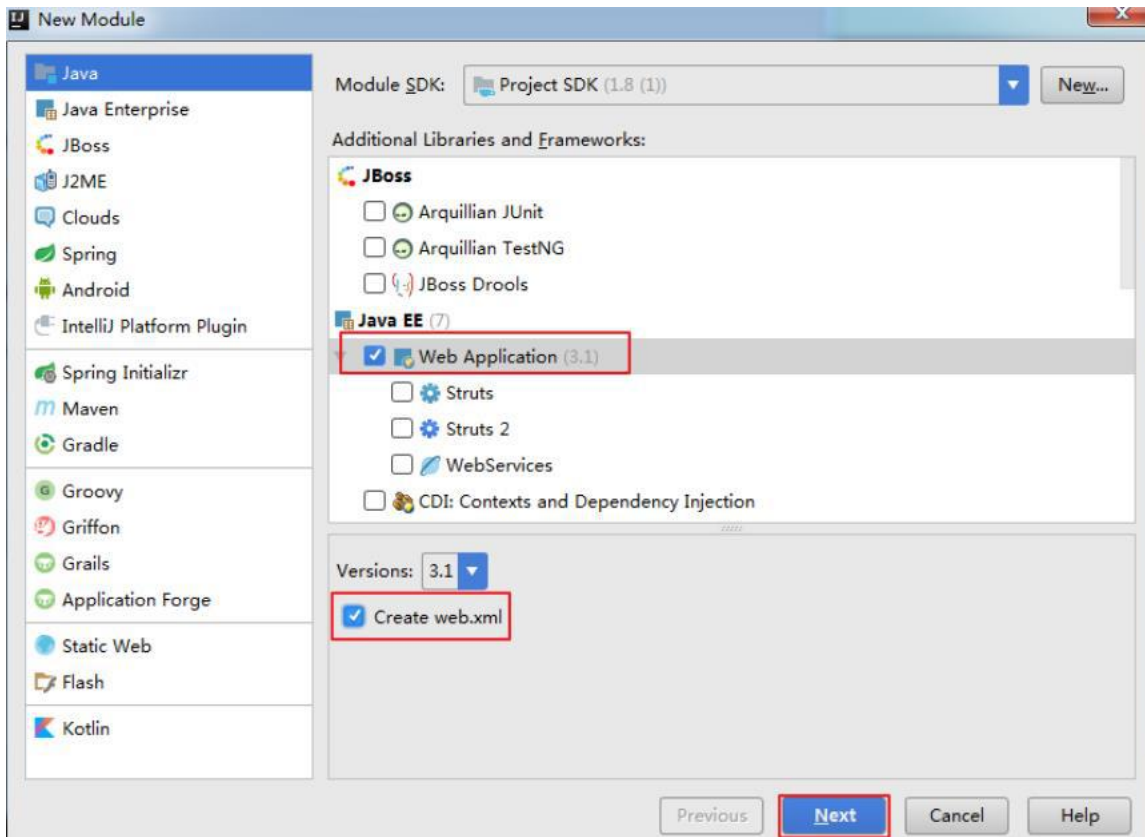
这里需要明确一下概念：

IDEA中的项目（**project**）与 eclipse中的项目（**project**）是不同的概念，IDEA的 **project**相当于之前 eclipse的 **workspace**，IDEA的 **Module**是相当于 eclipse的项目（**project**）。这个地方刚开始用的时候会很容易理不清它们之间的关系。从 Eclipse转过来的人总是下意识地要再同一个窗口管理 n个项目，这在 IntelliJ IDEA是无法得到。IntelliJ IDEA提供的体验是：一个

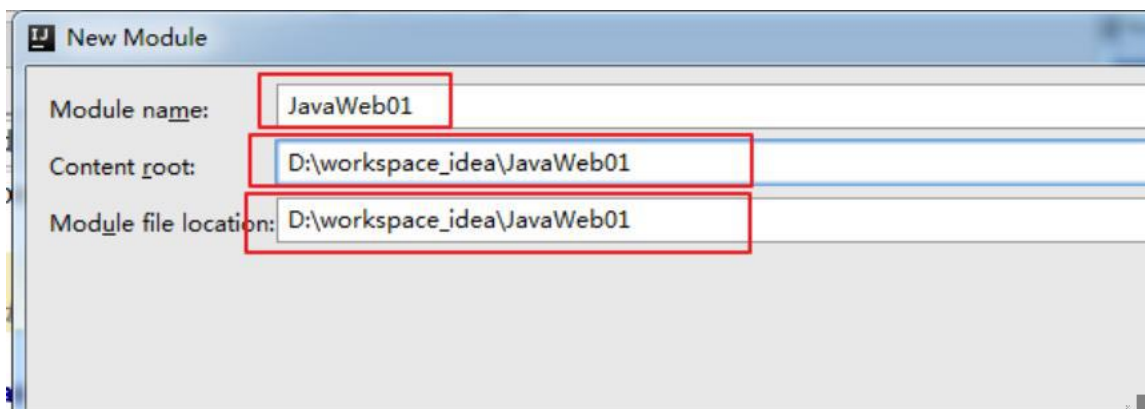


Project打开一个 Window窗口。

IDEA:project = Eclipse:Workspace ; IDEA:module = Eclipse:project

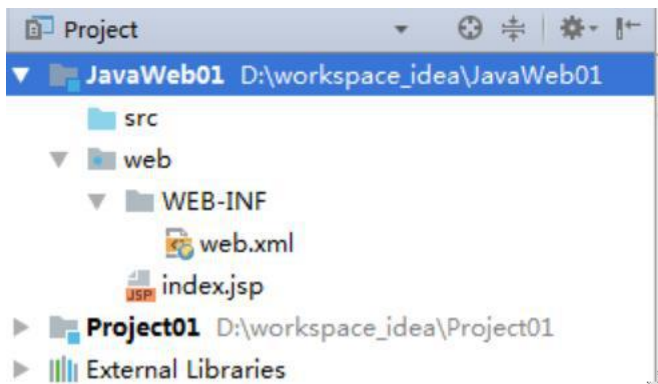


这里一定要勾选 Web Application，才能创建一个 Web工程。



提供 Web工程名。这里注意修改一下 Content root和 Module file location。

创建以后的工程结构如下：

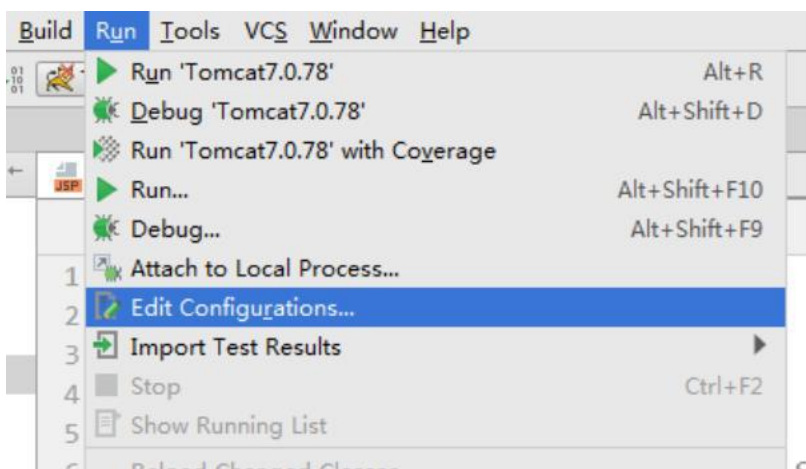


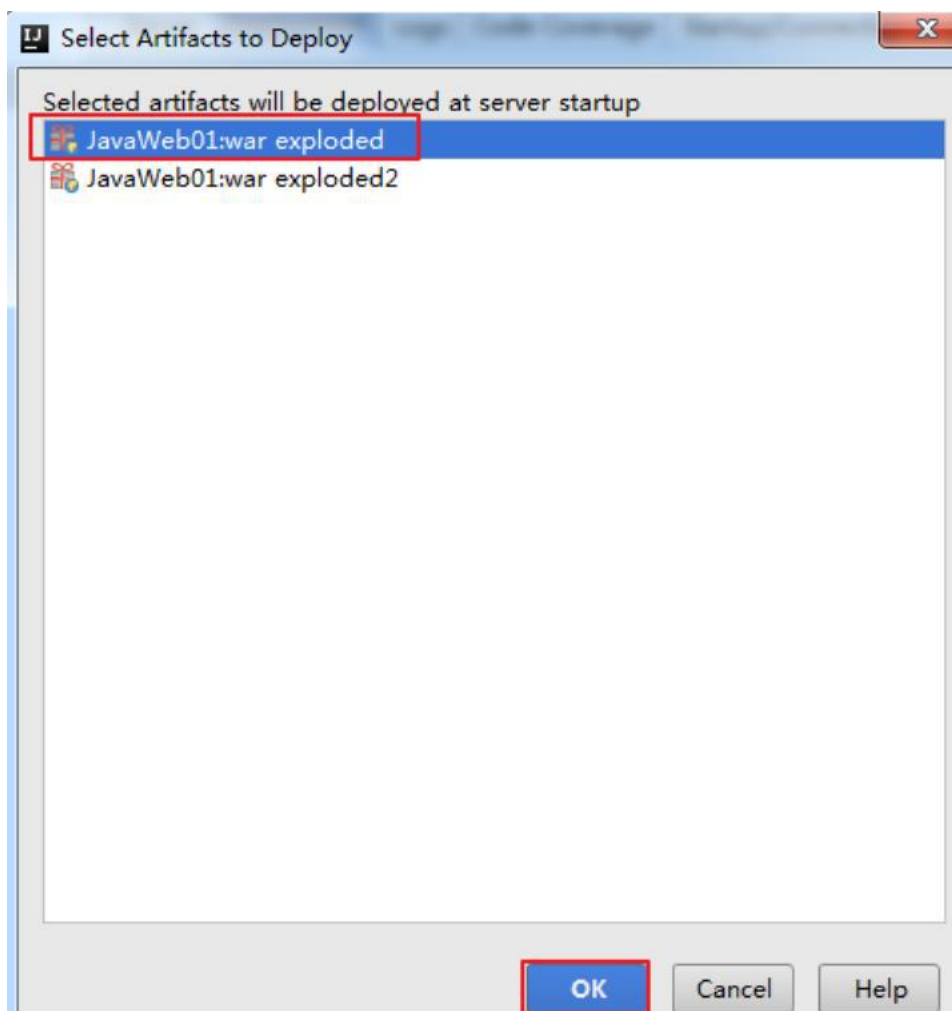
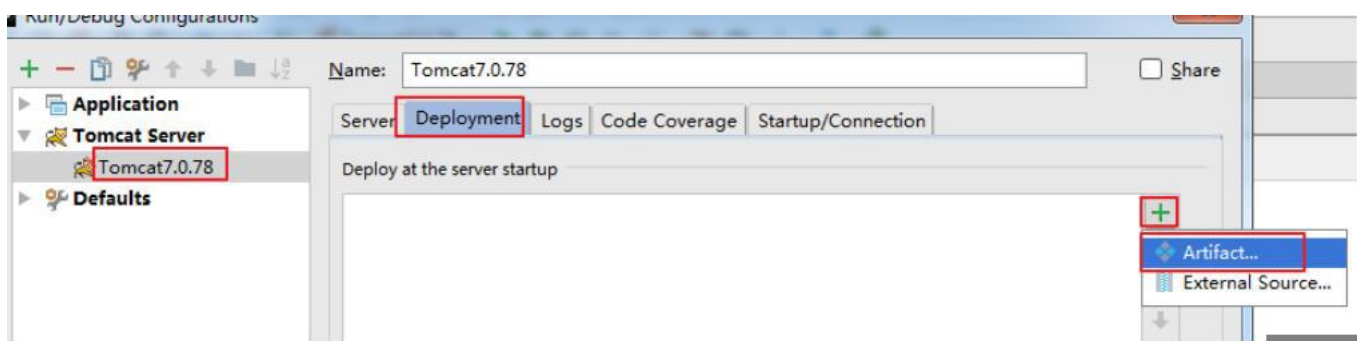
打开 index.jsp。修改为如下内容。这里你会发现 IDEA 的代码提示功能要强于 Eclipse。

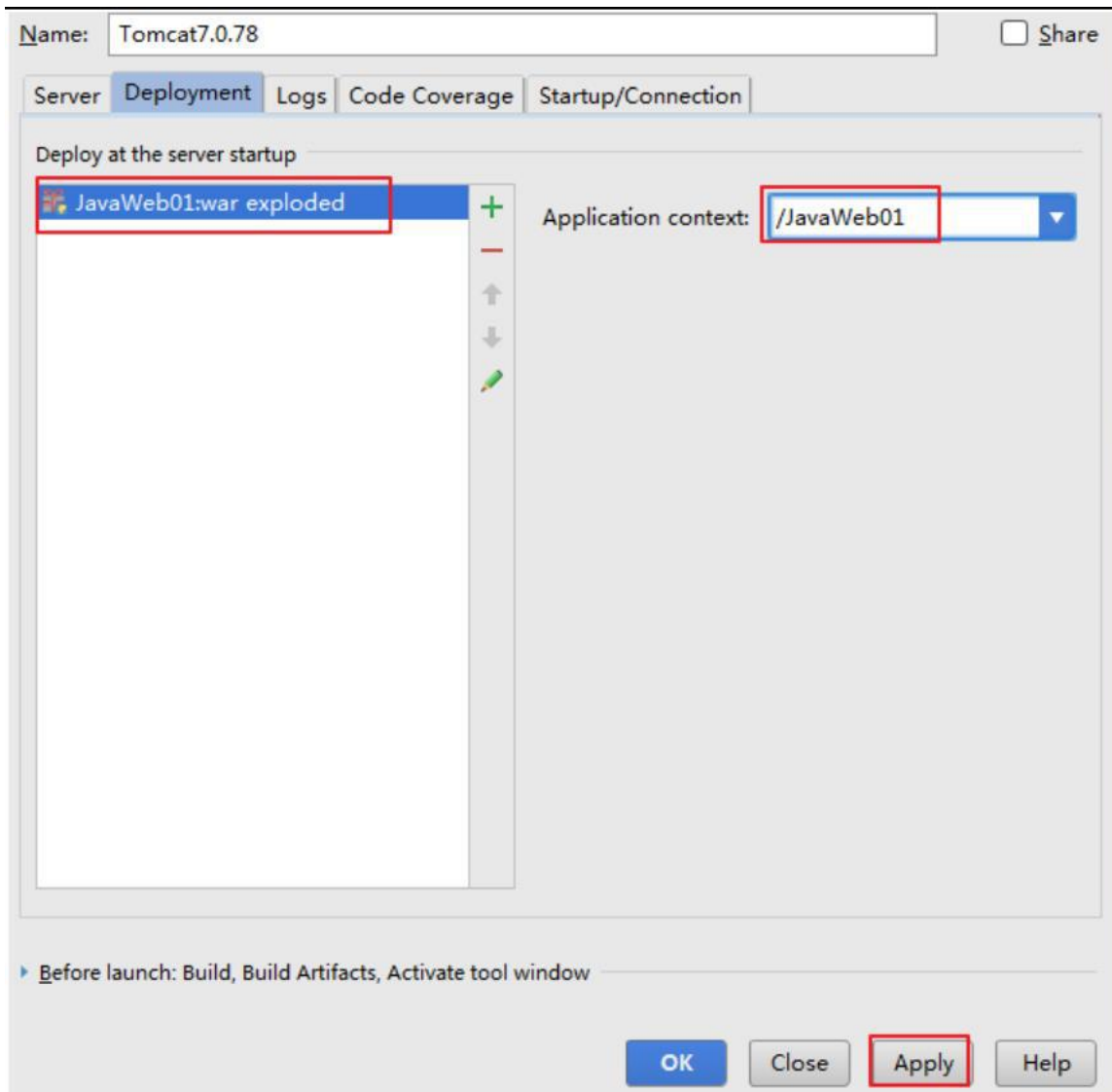
```
<%--
    Created by IntelliJ IDEA.
    User: 尚硅谷-宋红康
    Date: 2017/8/20 0020
    Time: 下午 11:43

    To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>尚硅谷官网</title>
</head>
<body>
    <h1 style="color: red">hello IDEA!</h1>
</body>
</html>
```

下面发布工程到 tomcat 中。发布的方式很多，这里说一种比较简洁的方式：

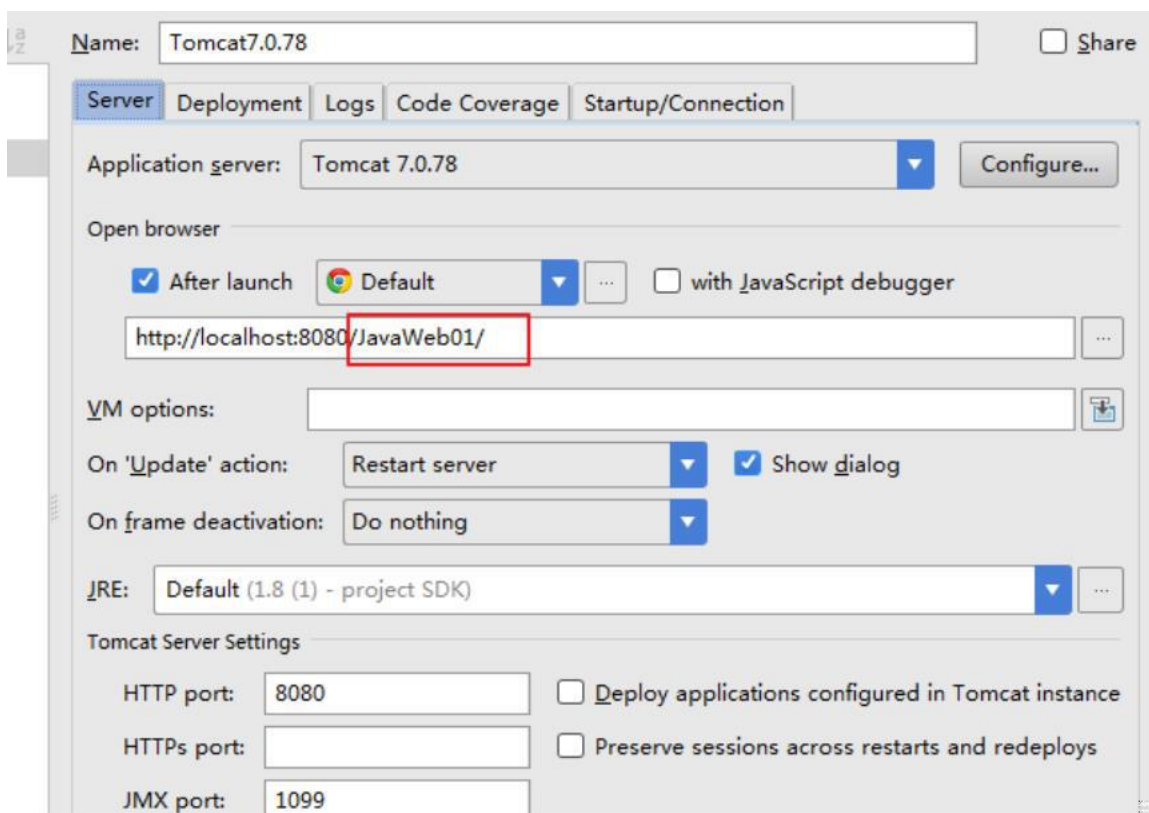




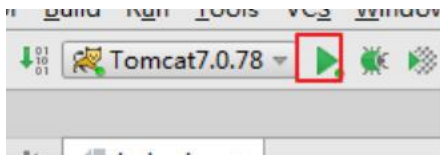


选择 Web工程，并提供应用名称，可以自定义。点击 apply

查看 Server项：其中显示了刚才设置的应用名。



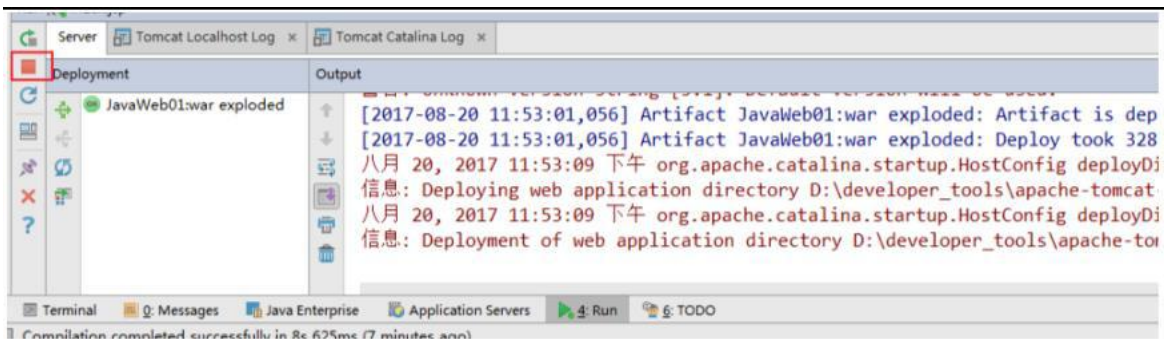
点击 OK以后，执行刚才创建的 index.jsp即可



# hello IDEA!

注意事项:

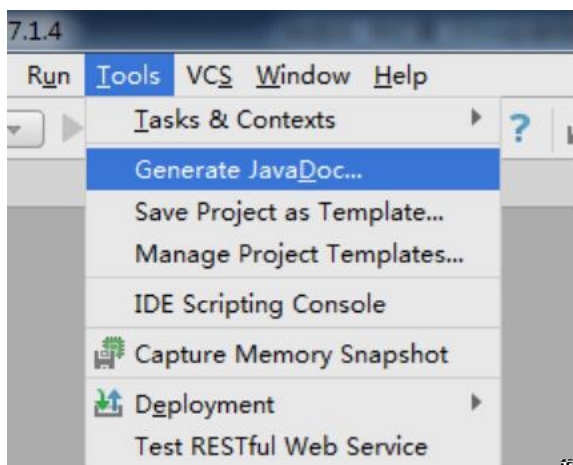
显示运行以后的 Tomcat的信息:

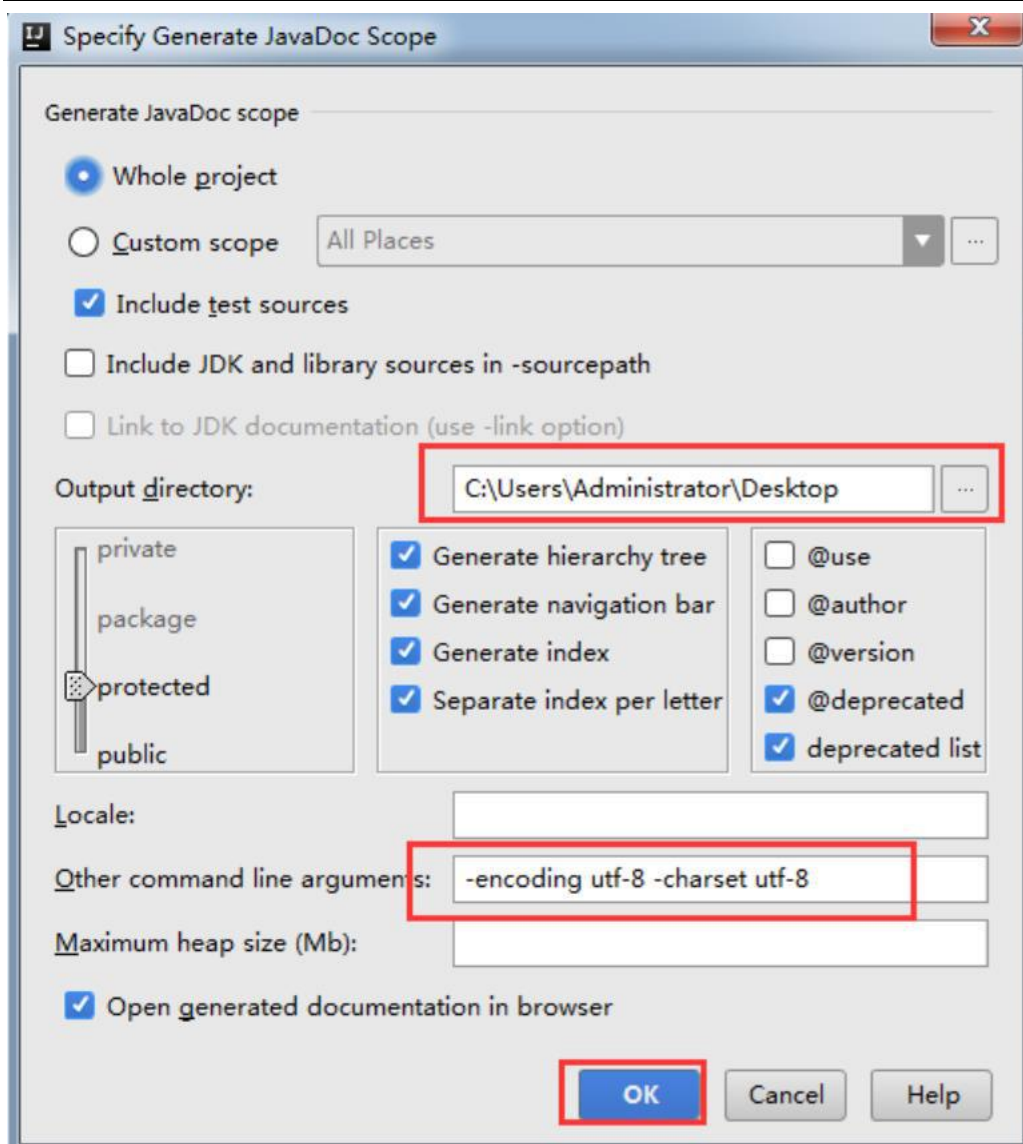


可以点击红框，注意点击一次并不能关闭服务器，只是断开了与服务器的连接，可以再点击一下，停止按钮显示为灰色，才表示关闭。

## 十、其它设置

### 1.生成 javadoc



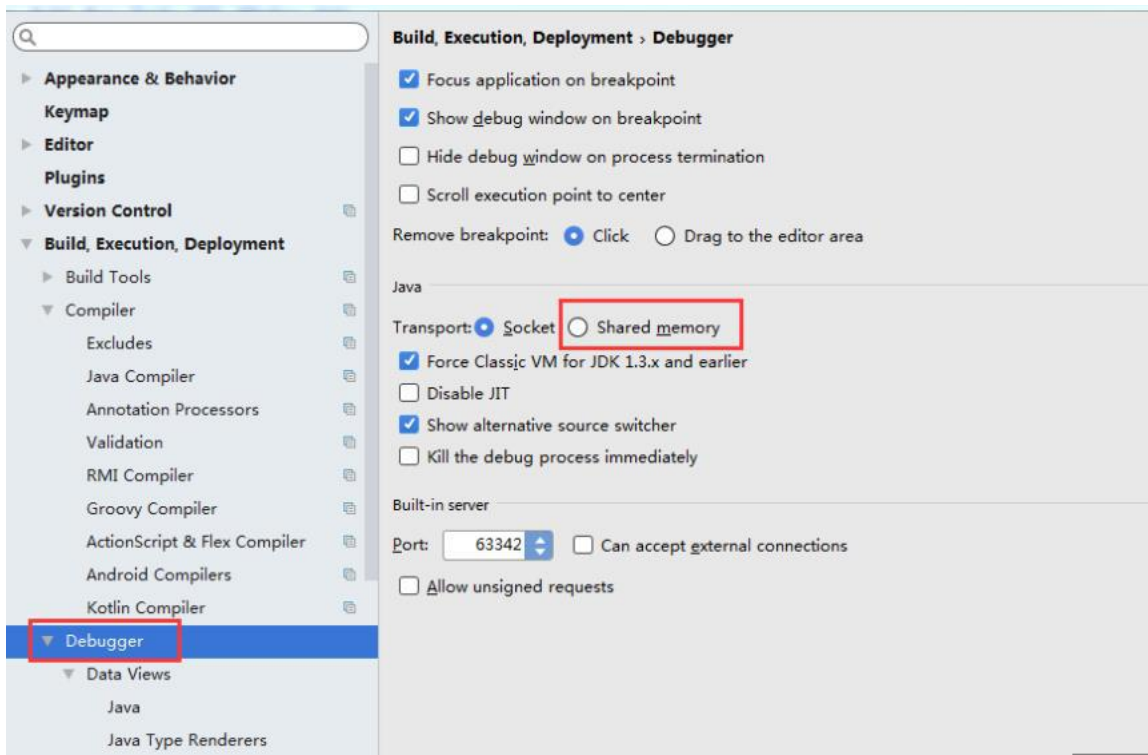


输入:

-encoding utf-8 -charset utf-8

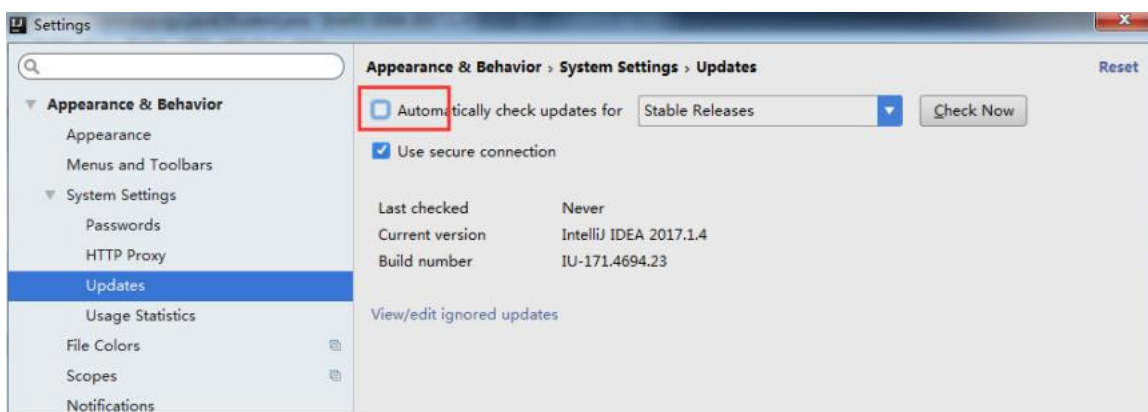


## 2. 设置运行内存的优化



选择红框 shared memory：选择此种方式运行，内存占用相对较少。

## 3. 取消更新

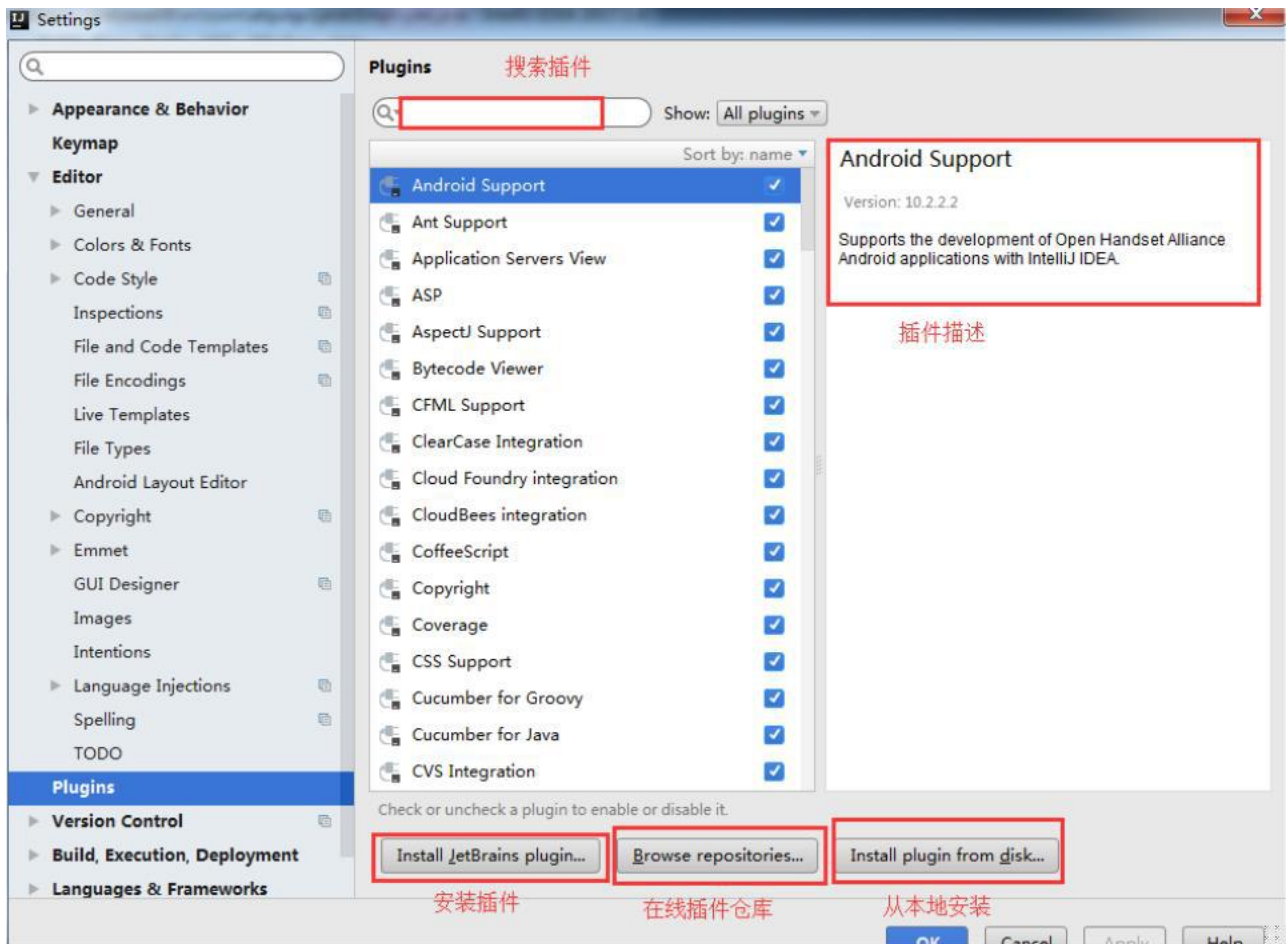


取消勾选：即可取消更新

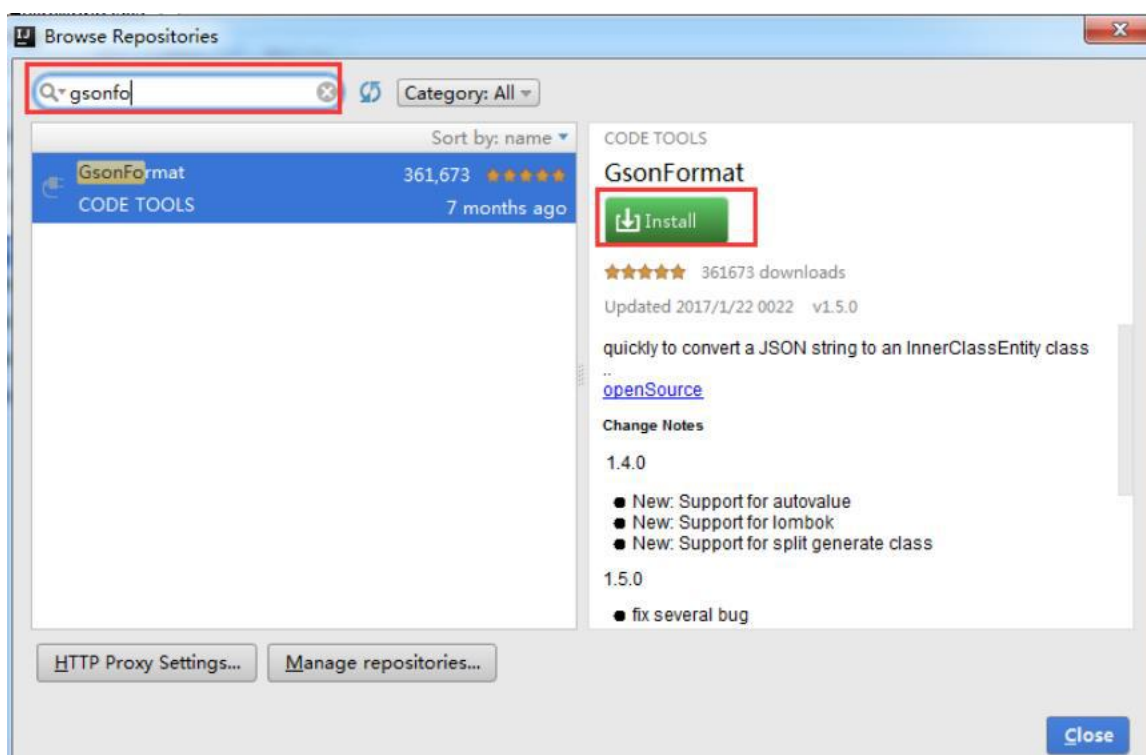
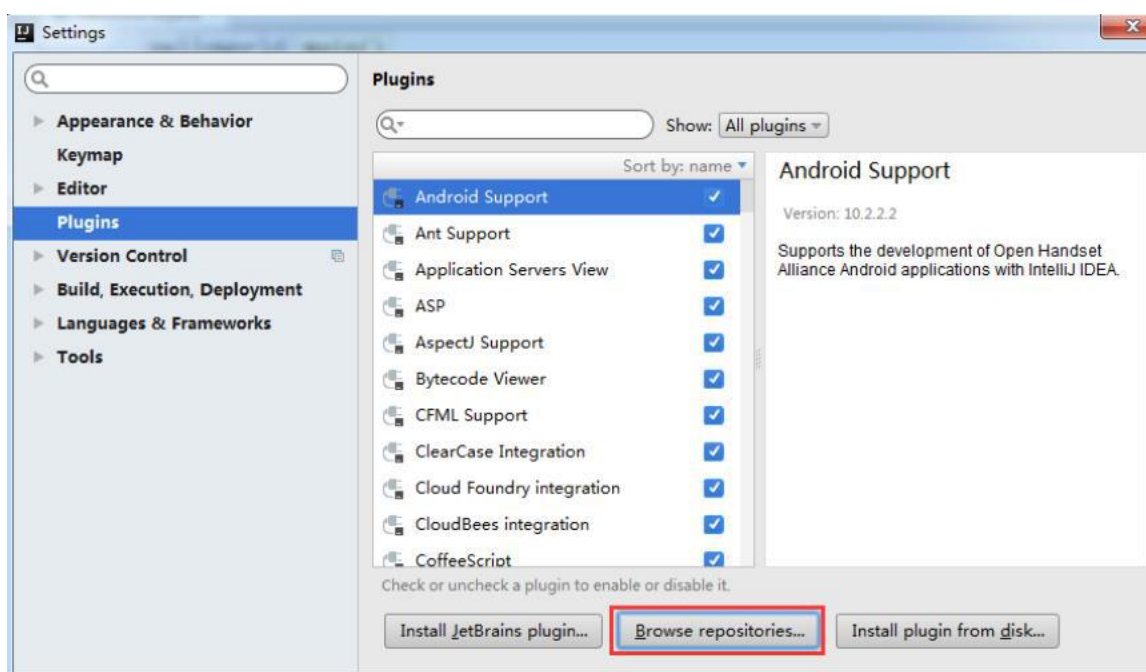
## 4. 安装在线和本地插件，取消插件启动

在 IntelliJ IDEA 的安装讲解中我们其实已经知道，IntelliJ IDEA 本身很多功能也都是通过插件的方式来实现的。

官网插件库：<https://plugins.jetbrains.com/>



需要特别注意的是：在国内的网络下，很经常出现显示不了插件列表，或是显示了插件列表，无法下载完成安装。这时候请自行开 VPN，一般都可以得到解决。



如上图演示，在线安装 IntelliJ IDEA插件库中的插件。

常用插件推荐：

插件名称	插件介绍	官网地址
------	------	------

插件名称	插件介绍	官网地址
Key promoter	快捷键提示	<a href="https://plugins.jetbrains.com/plugin/4455?pr=idea">https://plugins.jetbrains.com/plugin/4455?pr=idea</a>
CamelCase	驼峰式命名和下划线命名交替变化	<a href="https://plugins.jetbrains.com/plugin/7160?pr=idea">https://plugins.jetbrains.com/plugin/7160?pr=idea</a>
CheckStyle-IDEA	代码样式检查	<a href="https://plugins.jetbrains.com/plugin/1065?pr=idea">https://plugins.jetbrains.com/plugin/1065?pr=idea</a>
FindBugs-IDEA	代码 Bug检查	<a href="https://plugins.jetbrains.com/plugin/3847?pr=idea">https://plugins.jetbrains.com/plugin/3847?pr=idea</a>
Statistic	代码统计	<a href="https://plugins.jetbrains.com/plugin/4509?pr=idea">https://plugins.jetbrains.com/plugin/4509?pr=idea</a>
JRebel Plugin	热部署	<a href="https://plugins.jetbrains.com/plugin/?id=4441">https://plugins.jetbrains.com/plugin/?id=4441</a>
CodeGlance	在编辑代码最右侧，显示一块代码小地图	<a href="https://plugins.jetbrains.com/plugin/7275?pr=idea">https://plugins.jetbrains.com/plugin/7275?pr=idea</a>
Eclipse Code Formatter	使用 Eclipse的代码格式化风格，在一个团队中如果公司有规定格式化风格，这个可以使用。	<a href="https://plugins.jetbrains.com/plugin/6546?pr=idea">https://plugins.jetbrains.com/plugin/6546?pr=idea</a>
GsonFormat	把 JSON字符串直接实例化成类	<a href="https://plugins.jetbrains.com/plugin/7654?pr=idea">https://plugins.jetbrains.com/plugin/7654?pr=idea</a>

## 十一、版本控制(Version Control)

