1. IntelliJ IDEA版本

有两个IntelliJ IDEA的版本：社区版（IntelliJ IDEA Community Edition）和旗舰版（IntelliJ IDEA Ultimate Edition）。 社区版是免费​​的，开源的，但功能特性少一些。旗舰版是商业版本，但提供了一个优秀的工具集和功能。

1. JetBrains Toolbox

Toolbox 官网

在安装Intellij IDEA之前, 推荐大家安装JetBrains出品的工具箱, 该软件可以下

载,安装和更新JetBrains下的所有产品.也就是可以通过该软件下载IDEA和

WebStorm, 并在IDEA和WebStorm有更新时第一时间收到更新通知.

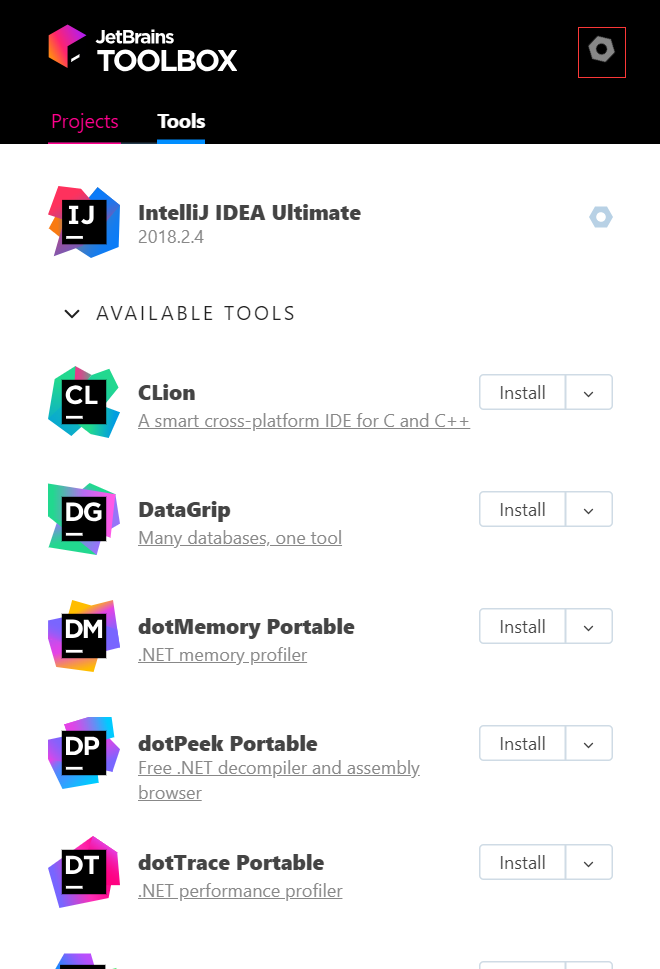
安装完Toolbox后, 在任务栏会有一个常驻图标, 由此打开后进入Tools页签,

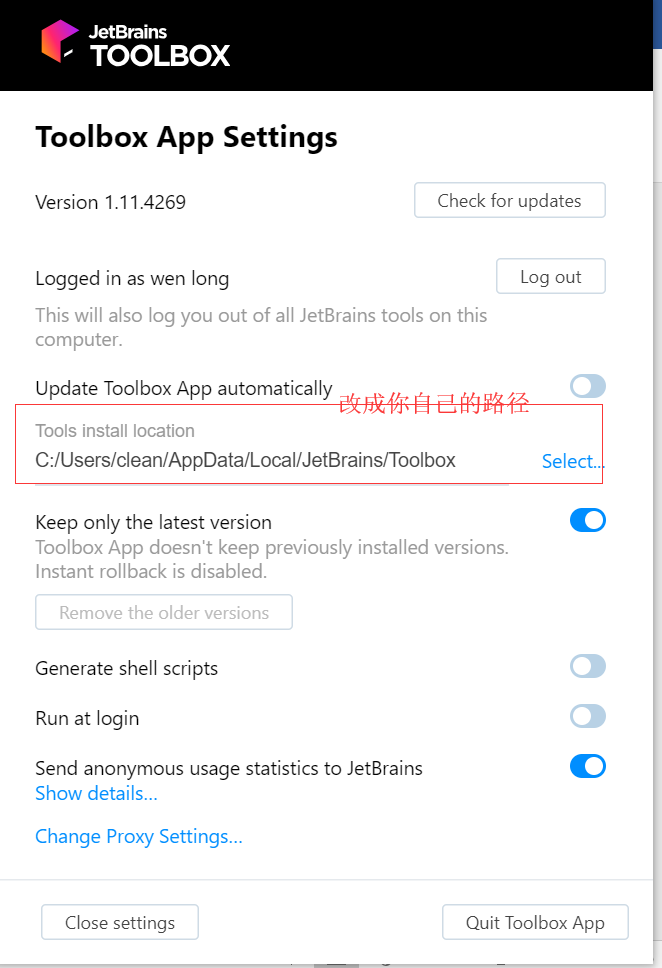
可以看到工具列表.注意要注册账号才可以。

在下载安装之前, 请先修改工具安装目录, 默认目录在C盘, 可能占用不必要的

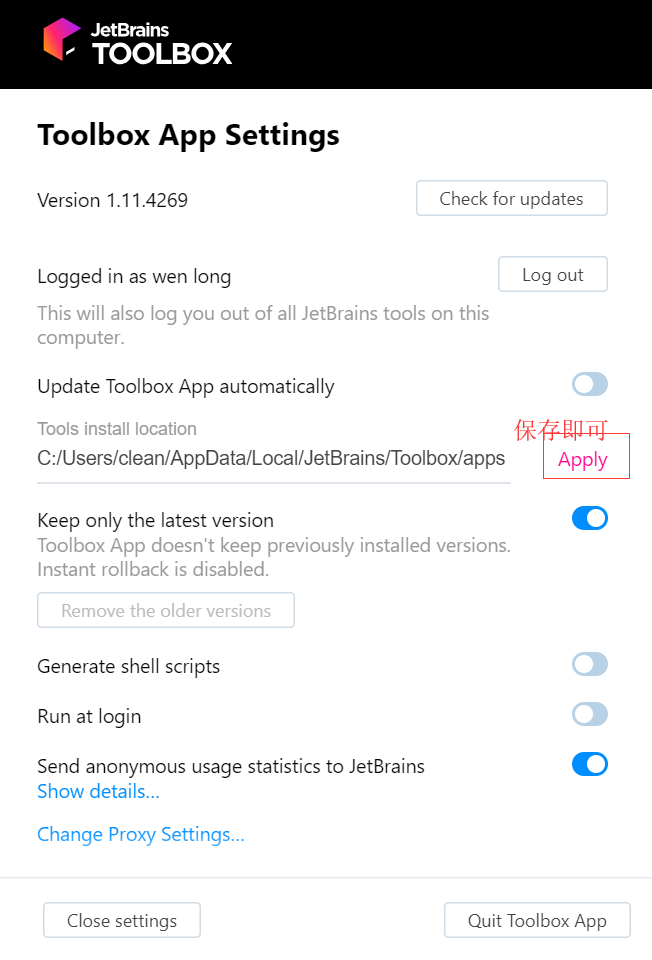
空间. 点击界面右上角的六边形齿轮, 点击settings, 找到install location, 修改

为你希望的安装目录, 再Apply changes即可.

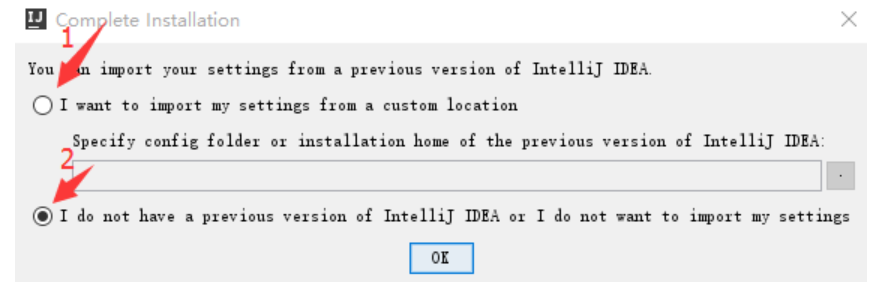




之后你便可以安装你喜欢的工具了, 直接点击工具右侧的install即可.

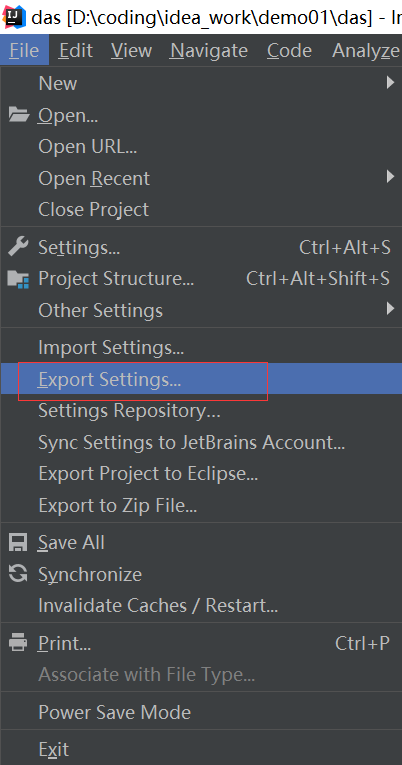


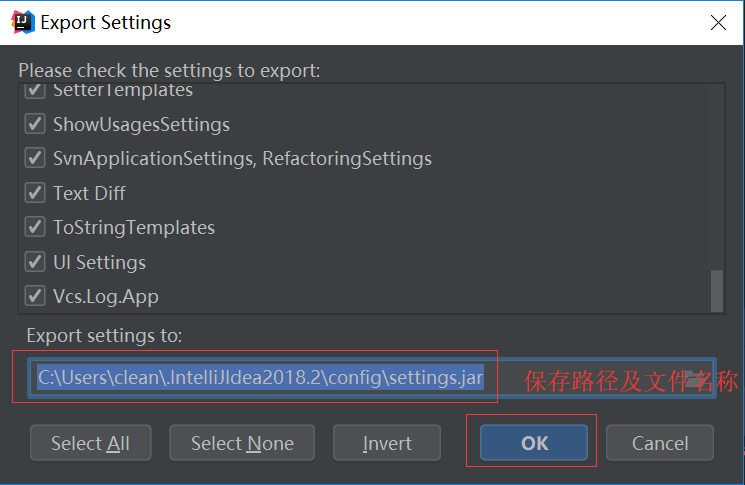
1. 破解idea ultimate
2. 导入已有的idea配置
3. 第一次启动时



标注 1：如果我们的电脑曾经安装过 IntelliJ IDEA 并且保留了其配置文件目录的话，则可以通过此按钮进行导入； 标注 2：如果我们的电脑没有安装过早期任何版本的 IntelliJ IDEA 的话，则可以通过此按钮让 IntelliJ IDEA 生成一份新的配置文件。 此外，如果我们的电脑不是首次安装 IntelliJ IDEA 的话，则还会在 标注 1 的上面多出现一个按钮选项，为： I want to import my settings from a previous version. 其功能为自动识别曾经安装过的 IntelliJ IDEA 旧版本的配置，如果我们选择了此按钮，则 IntelliJ IDEA 将自动把旧版本的配置文件转移到新版本的配置文件目录中。

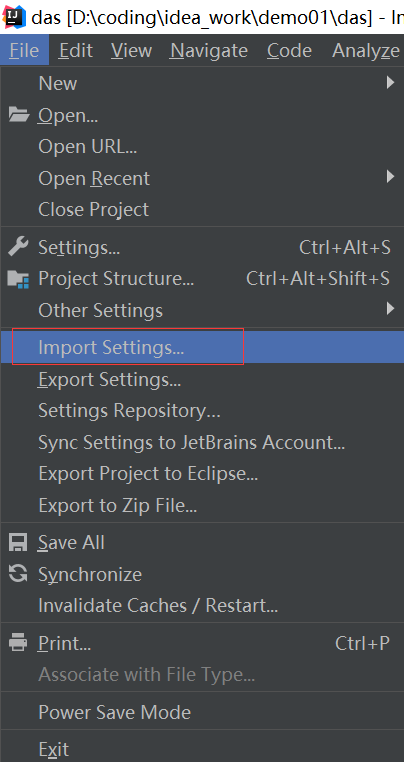
1. 导出已有的setting

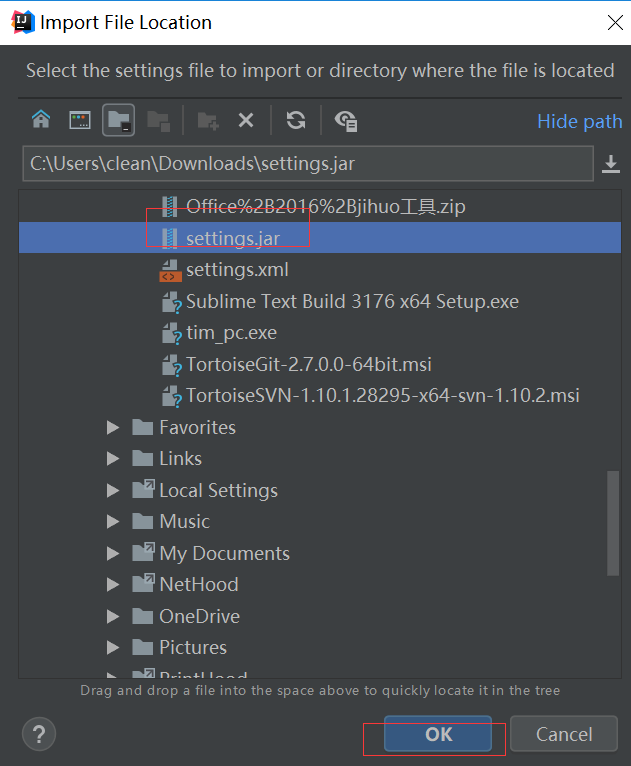




1. 导入已有的setting.jar。入口好几个，安装成功后第一次运行时，在弹出界面的右下角有settings入口，或者直接导入刚才导出的settings.jar文件。

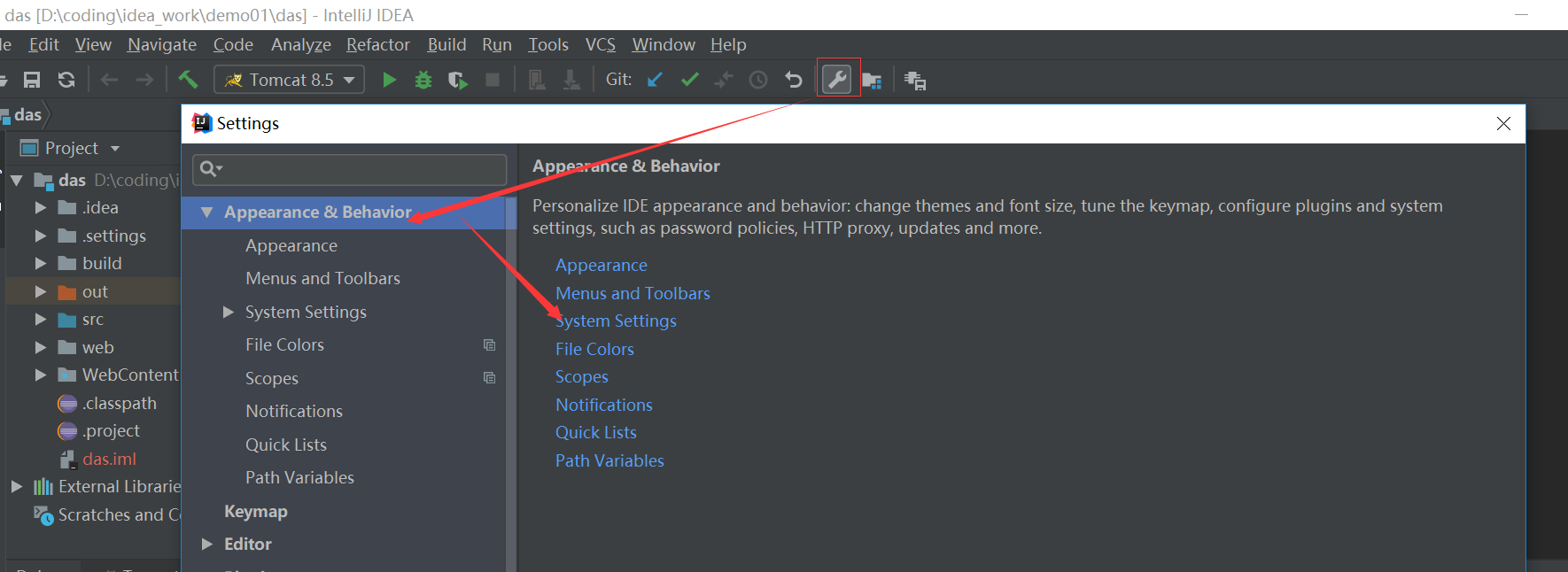
或者从以下方式进行导入

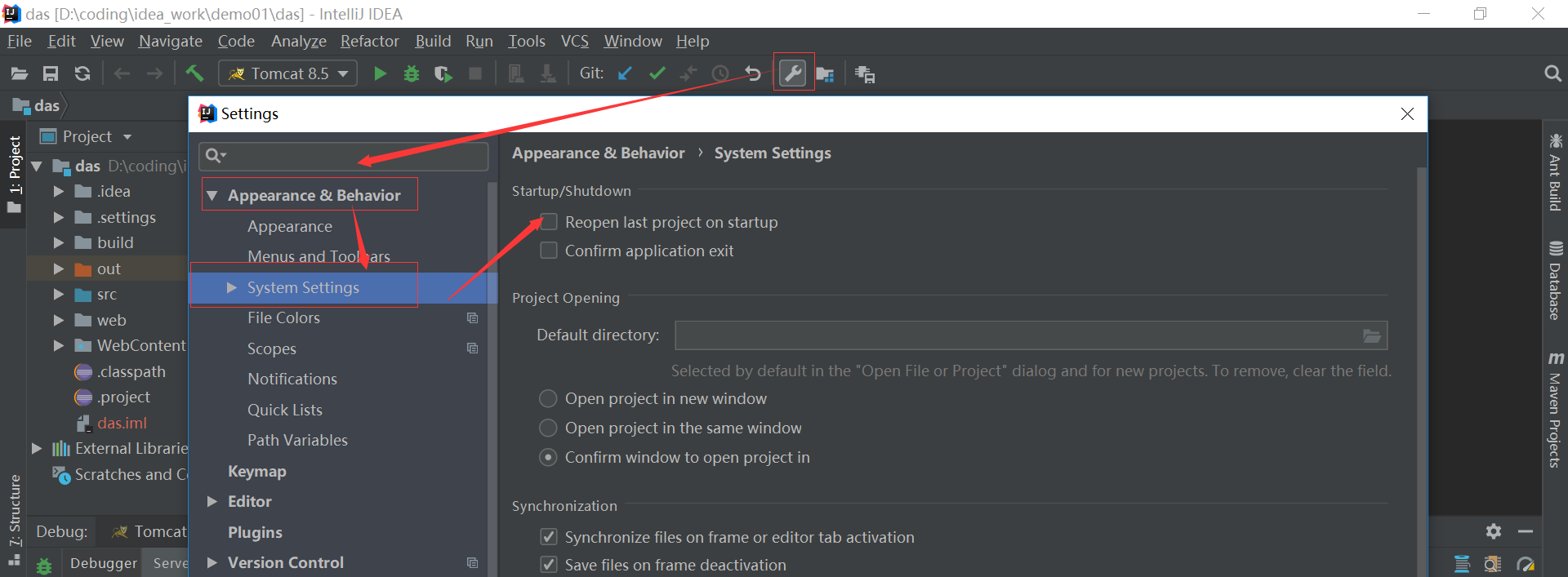




1. Idea常用配置
2. 启动后不要打开上一次的项目

找到Appearance & Behavior -> System Settings -> Reopen last project on startup取消该项的勾选

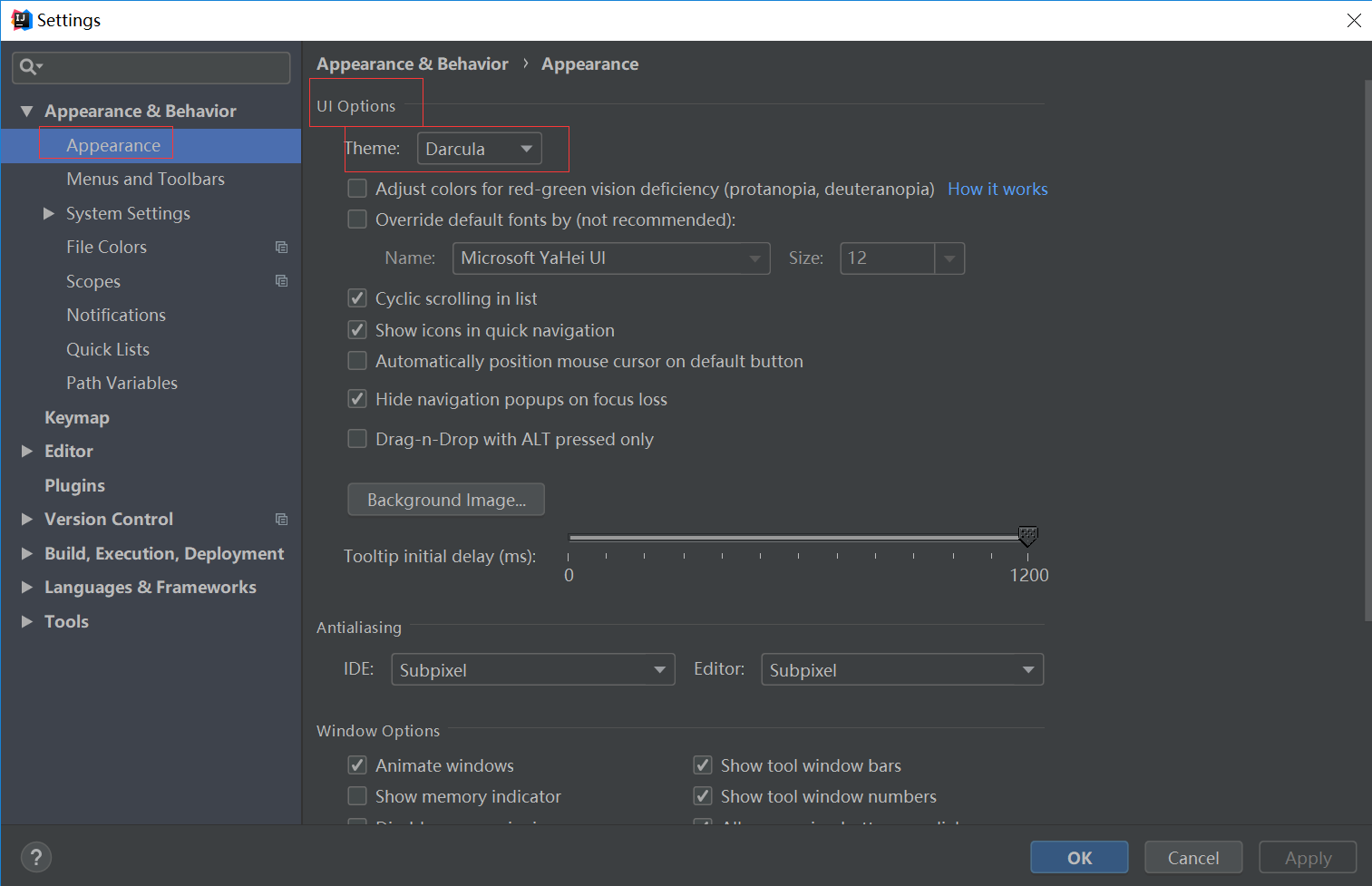




1. 界面风格与文字字体

界面主题推荐使用黑色界面风格Darcula, 亮度低护眼, 警告或提示也更醒目。

配置位置Appearance & Behavior -> Appearance -> UI Options -> Theme

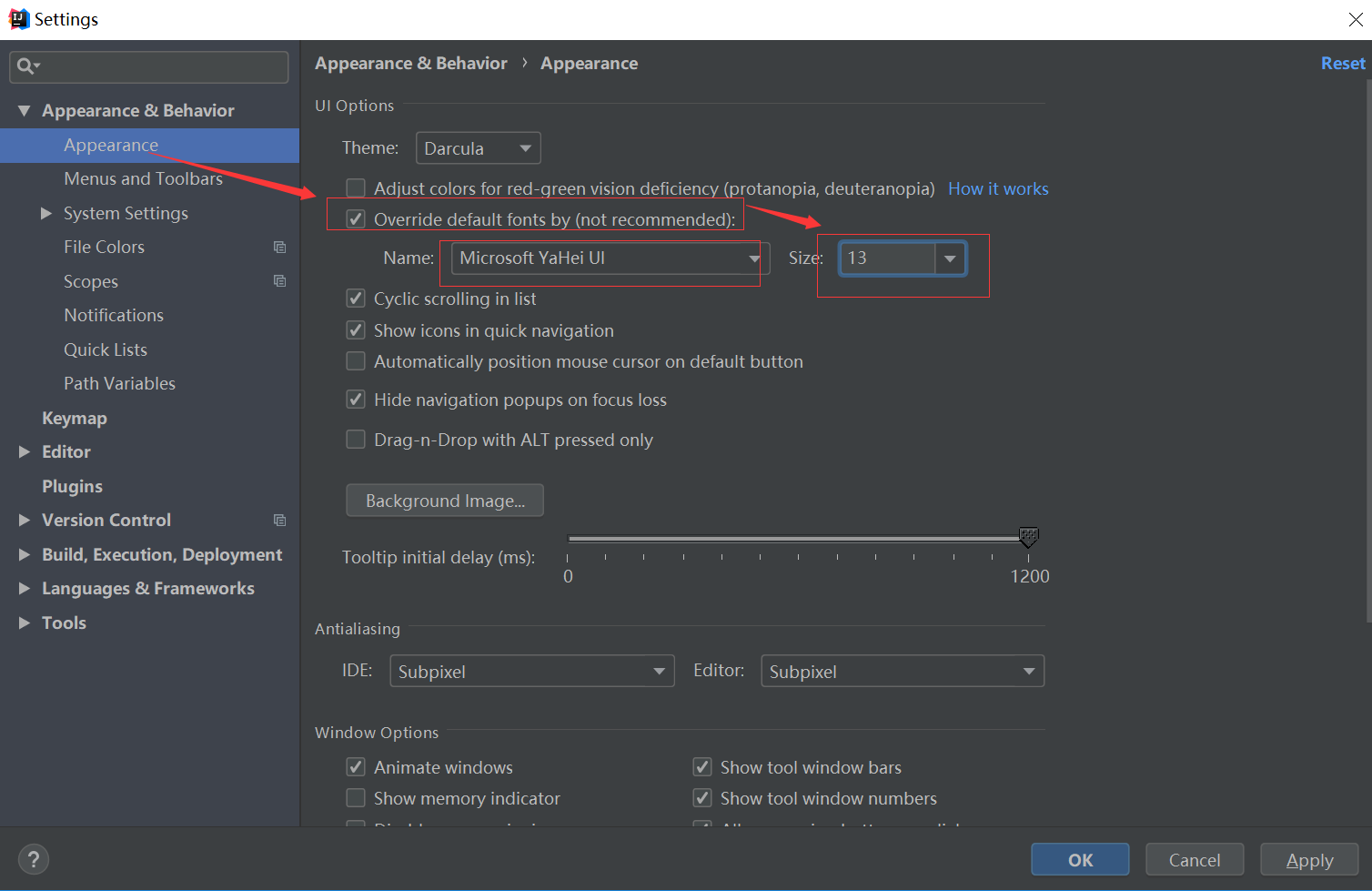


UI界面字体你可以根据喜好自己配置, 推荐微软雅黑

配置位置Appearance & Behavior -> Appearance -> UI Options -> Override

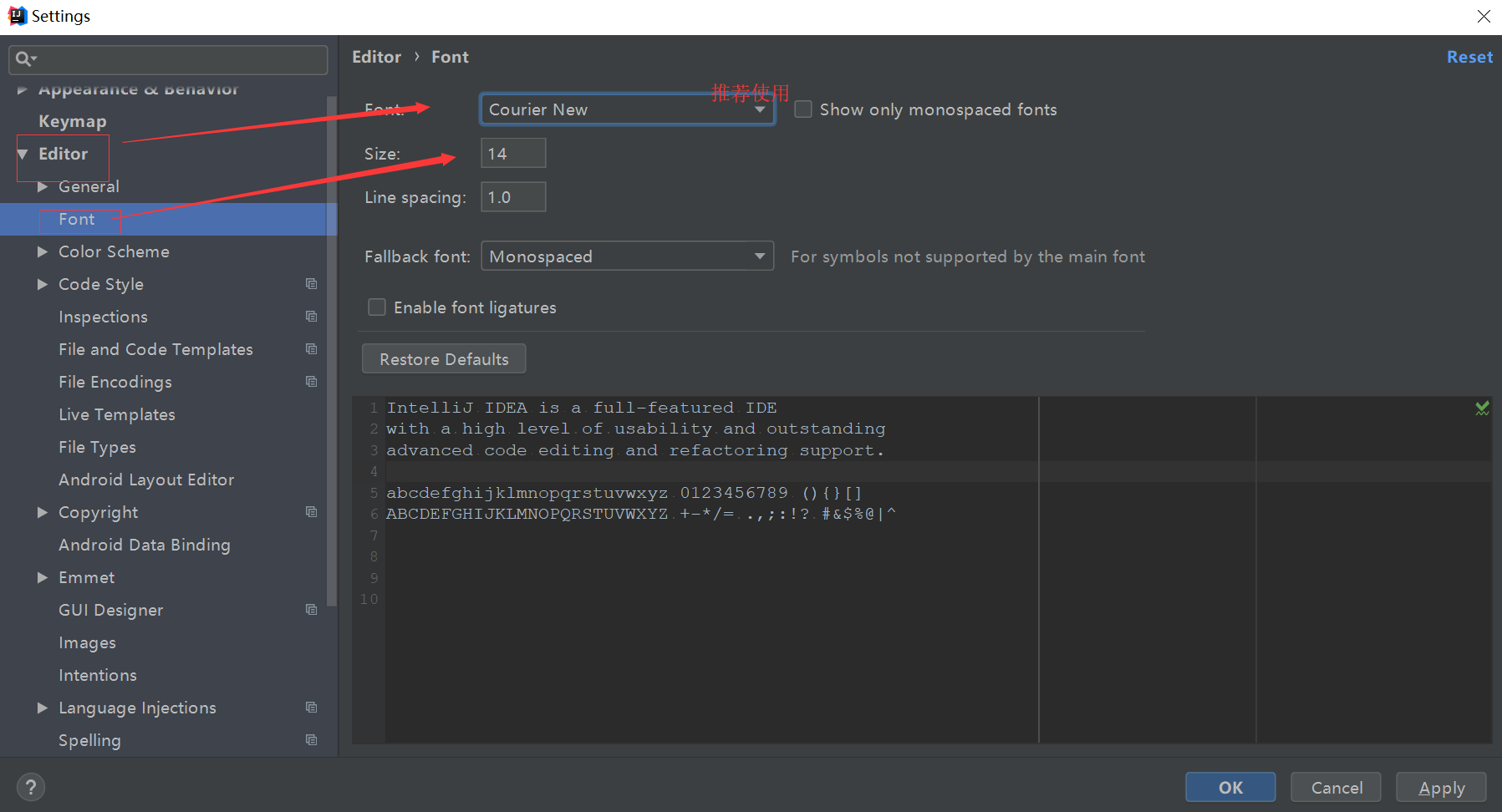
default fonts by (not recommended)

勾选Override default fonts by, 并配置Name为Microsoft YaHei UI, Size为13



编辑区域字体：

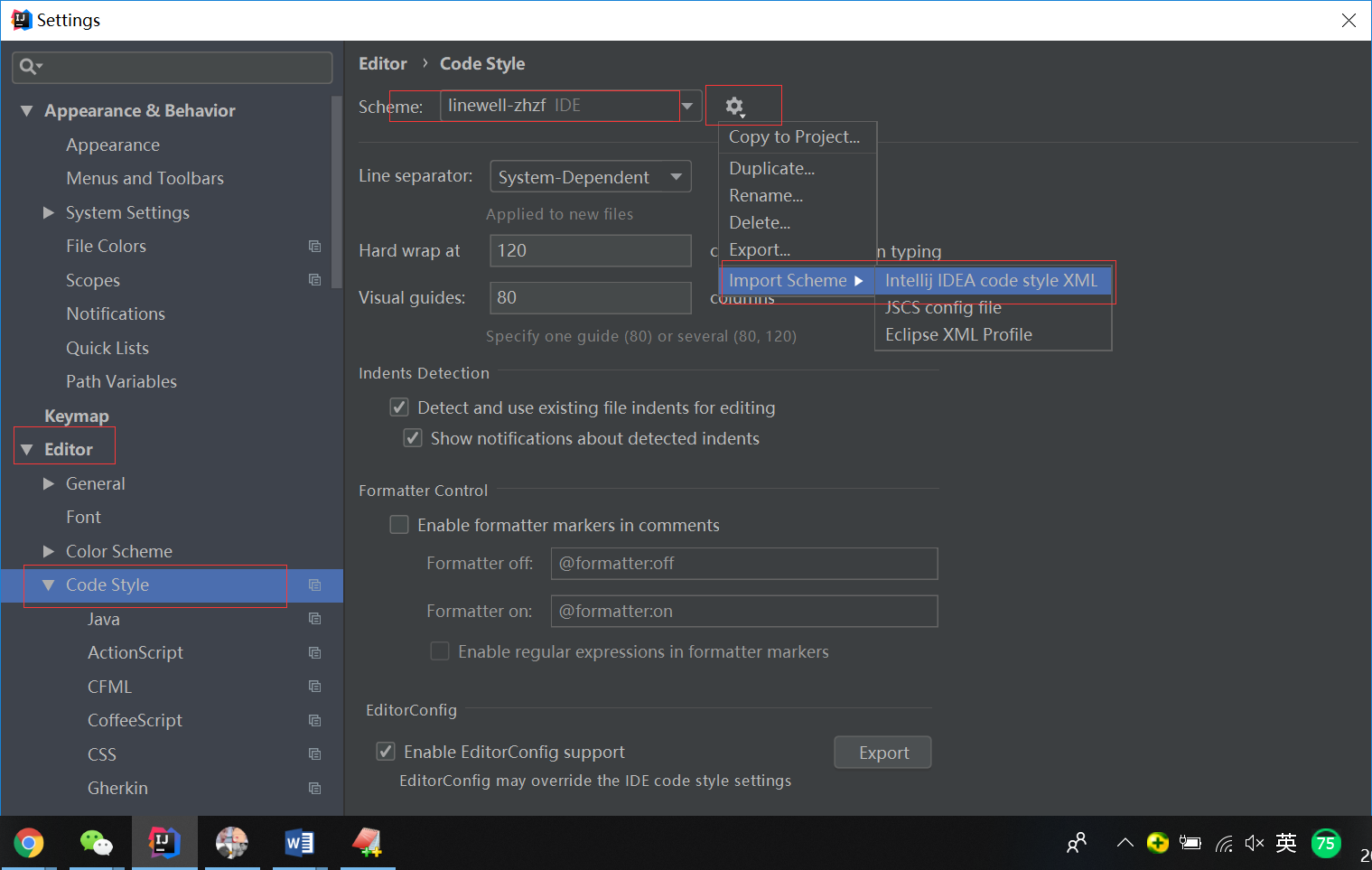
Editor -> Font, 推荐使用Courier New

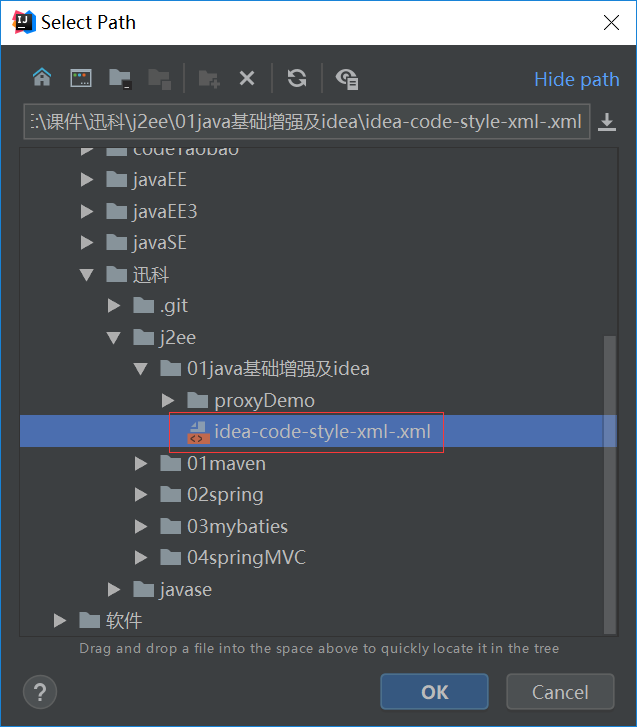


1. Code style

Editor -> Code Style -> Scheme点击Scheme下拉框右侧的齿轮, 选择Import Scheme -> Intellij IDEA code stype XML



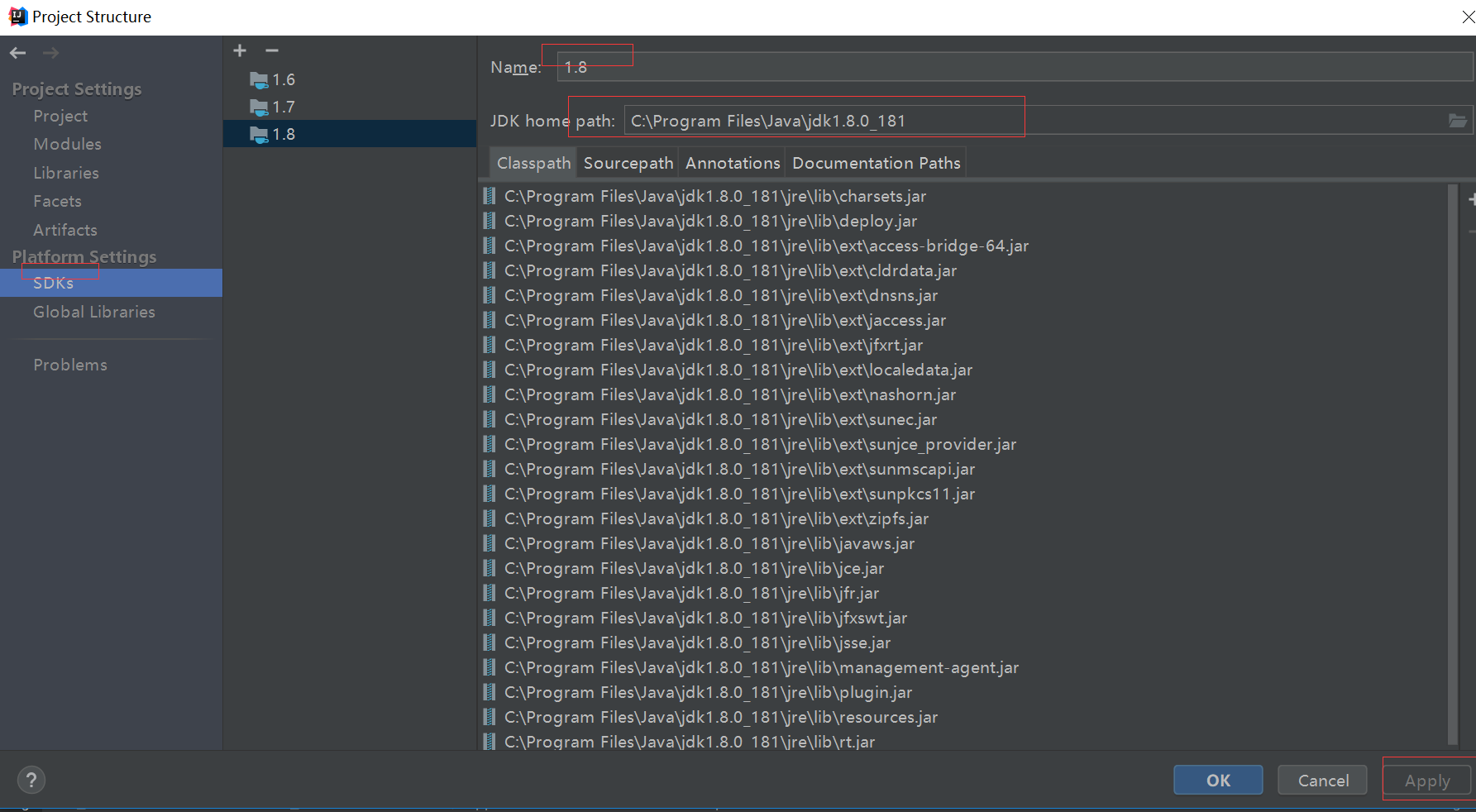




1. 配置jdk

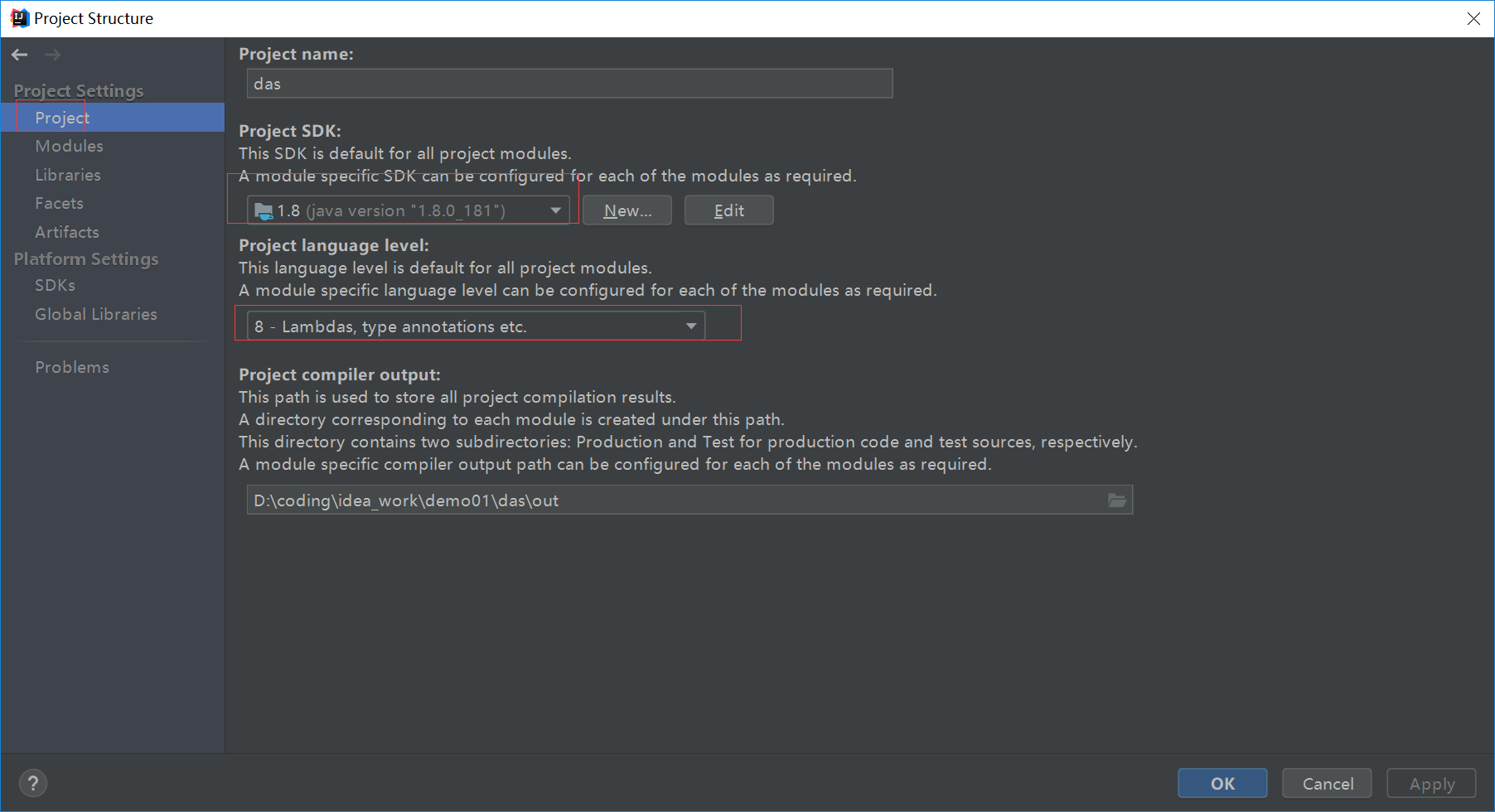
idea对jdk的配置包括设置工程的jdk（全局默认设置），设置工程语言级别，设置模块jdk，设置模块语言级别。你可以自己下载多个jdk，对不同项目设置不同的jdk

1. 设置jdk  file->project Structure ->sdks 点击加号选择jdk,然后选择自己的jdk安装目录即可，然后点击apply



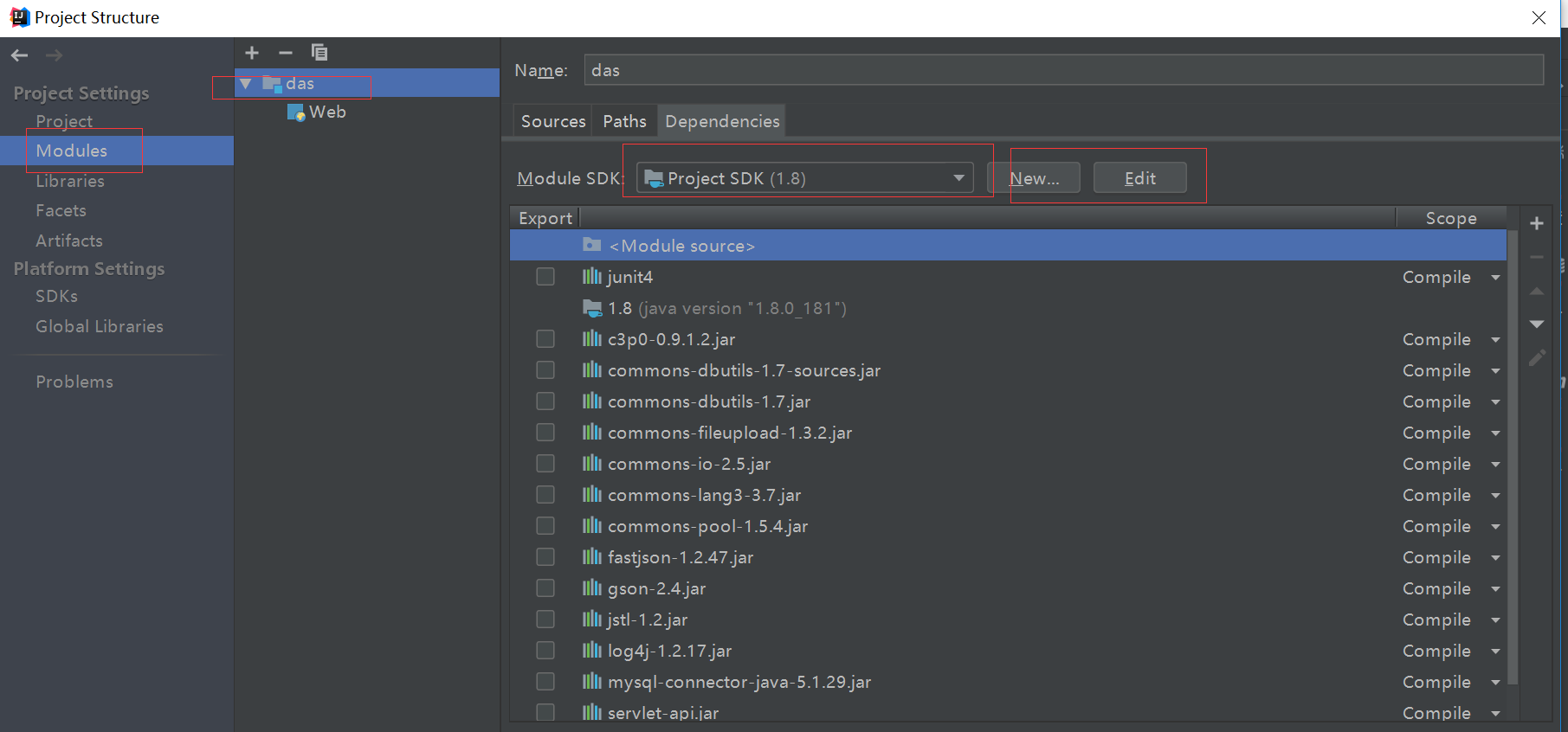
1. 设置project的jdk

project->project sdk 选择自己刚刚设置的sdk即可，然后点击apply即可完成jdk的设置。

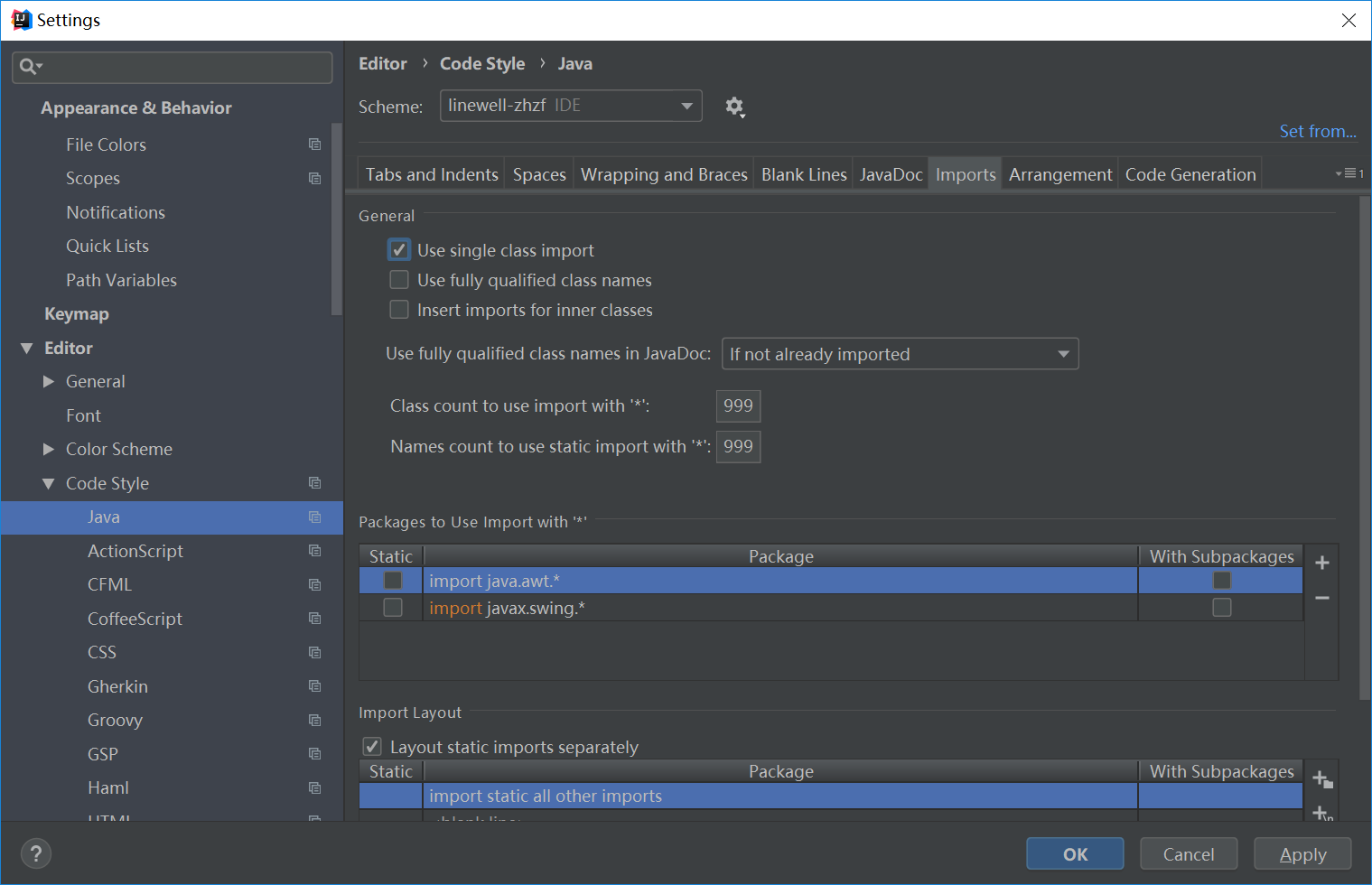


上图：配置JDK安装目录和工程的语言级别

1. 在Modules中可以单独设置语言级别

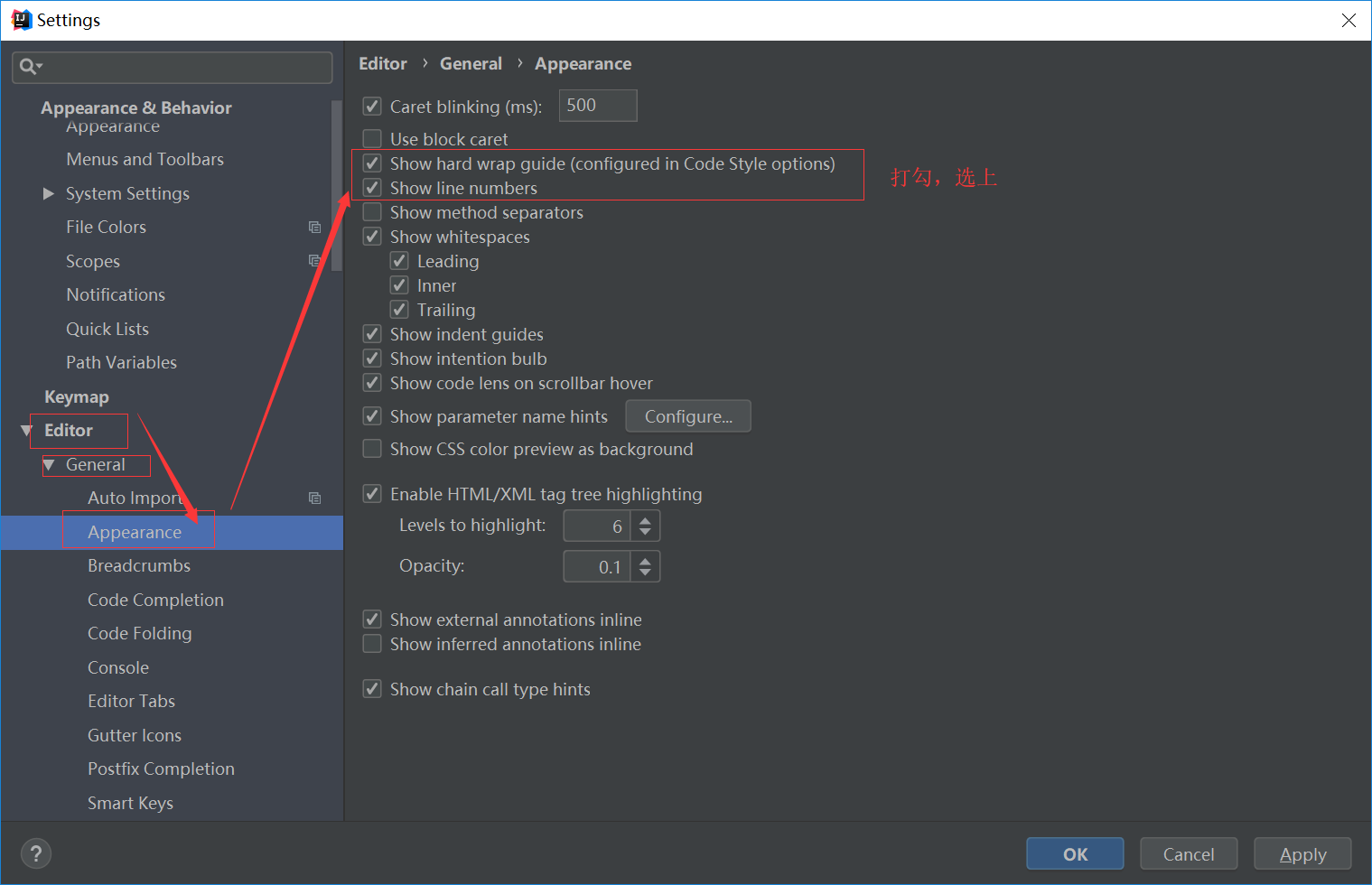


1. import语句顺序配置



1. 显示空格和换行

Editor -> General -> Appearance -> Show whitespaces将Show whitespaces和之后的 Leading, Inner, Trailling 的打钩都勾上。

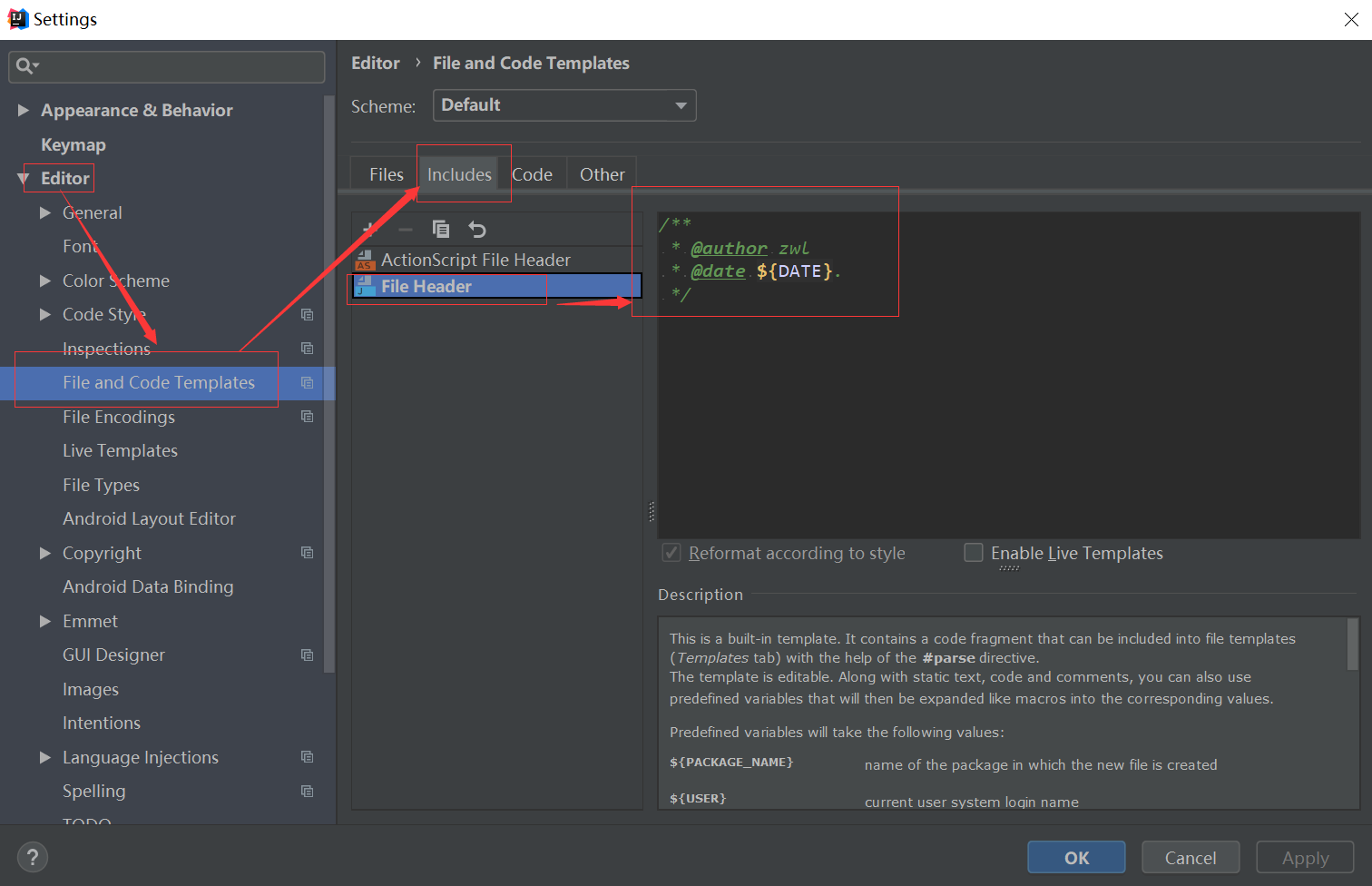


1. 代码提示修改为大小写不敏感(自动提示不限制首字母大小写)

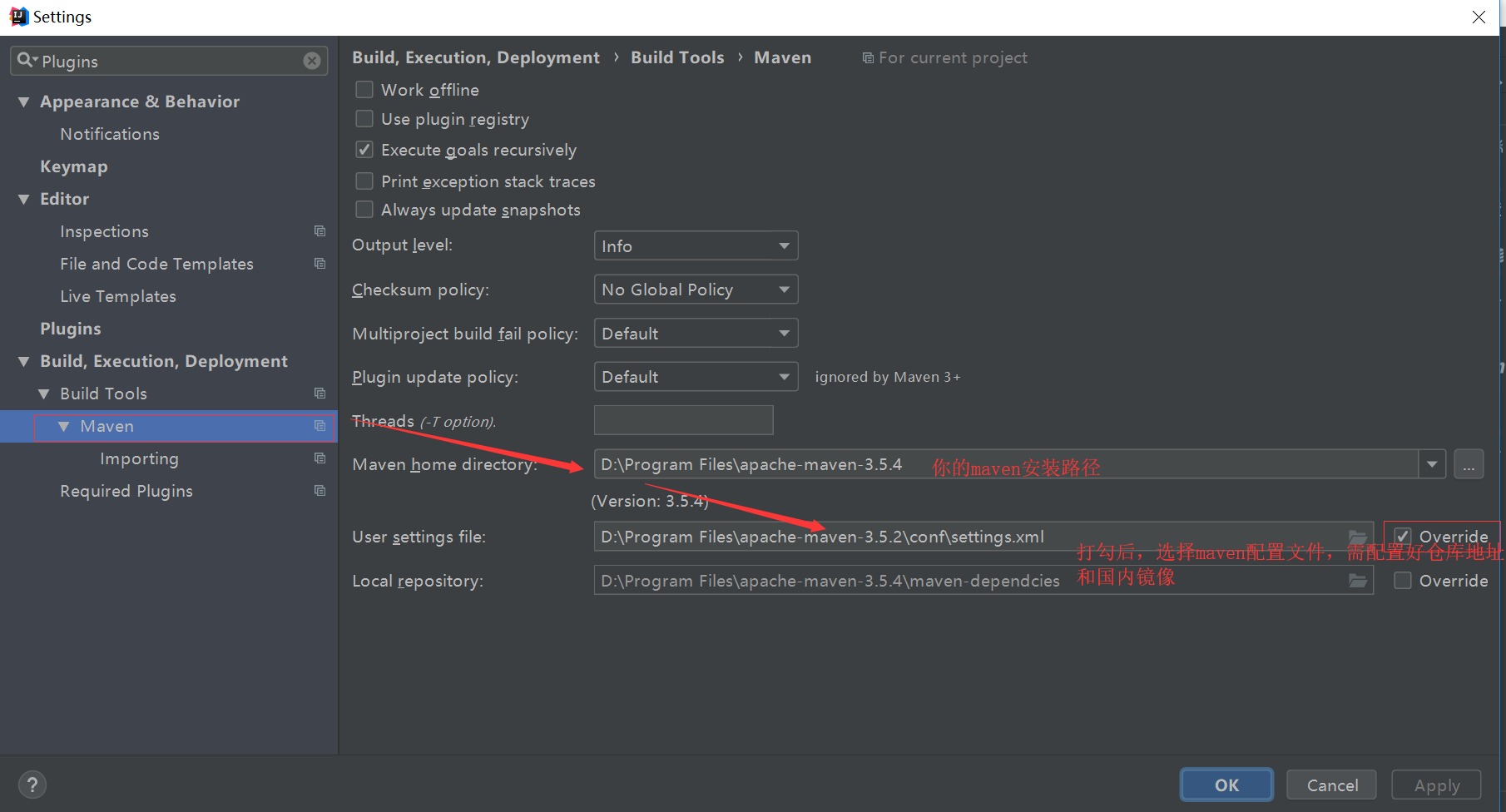
Editor=>Code Completion里把Case sensitive completion设置为None

2018.2.4版本貌似没有了。

1. 默认的类的Javadoc

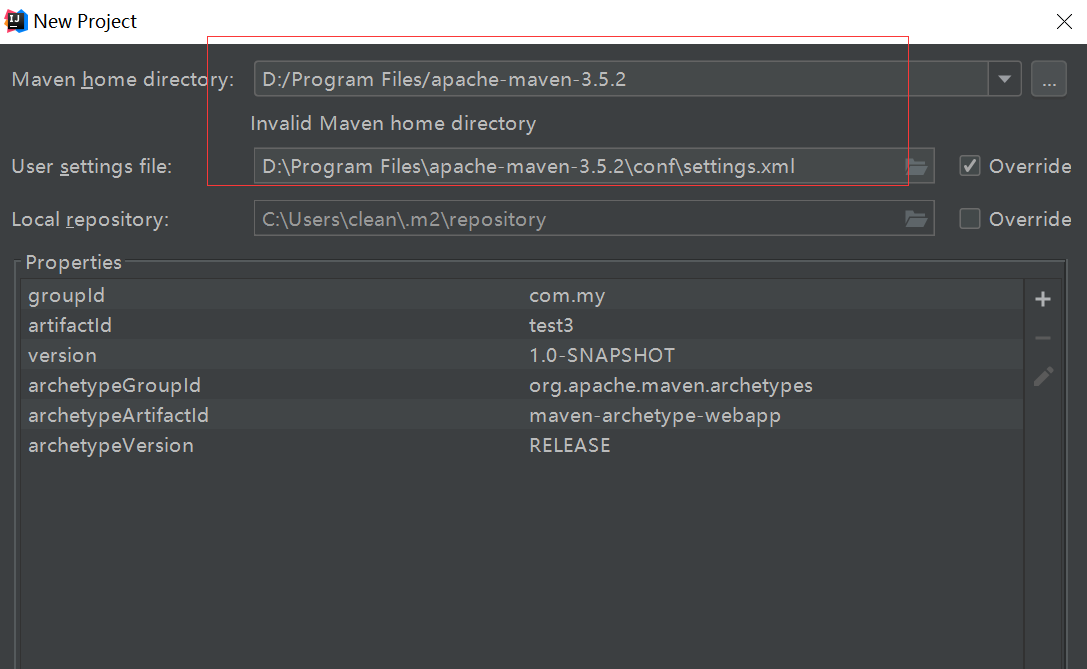


1. Maven配置

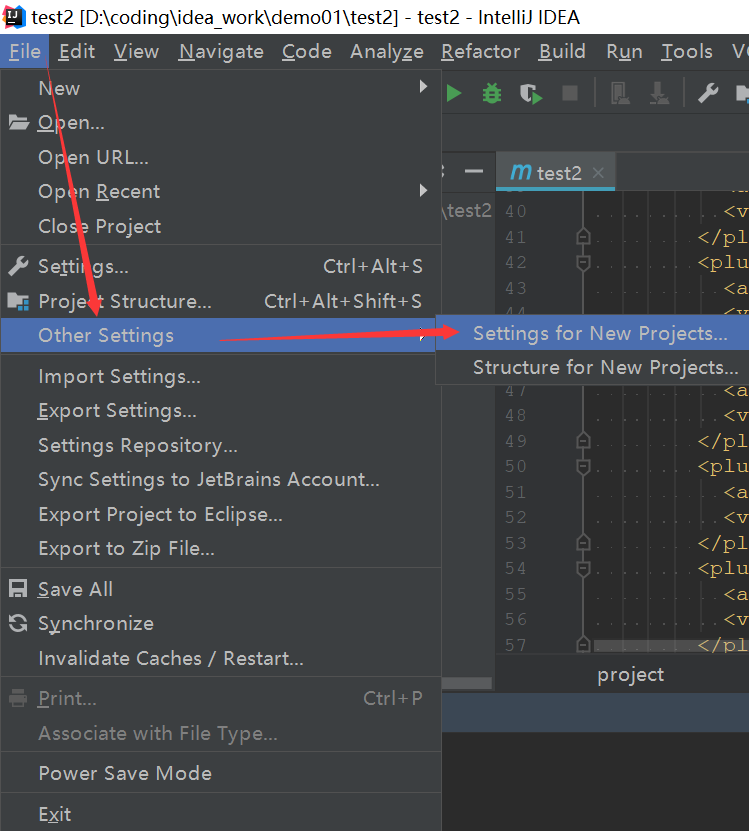


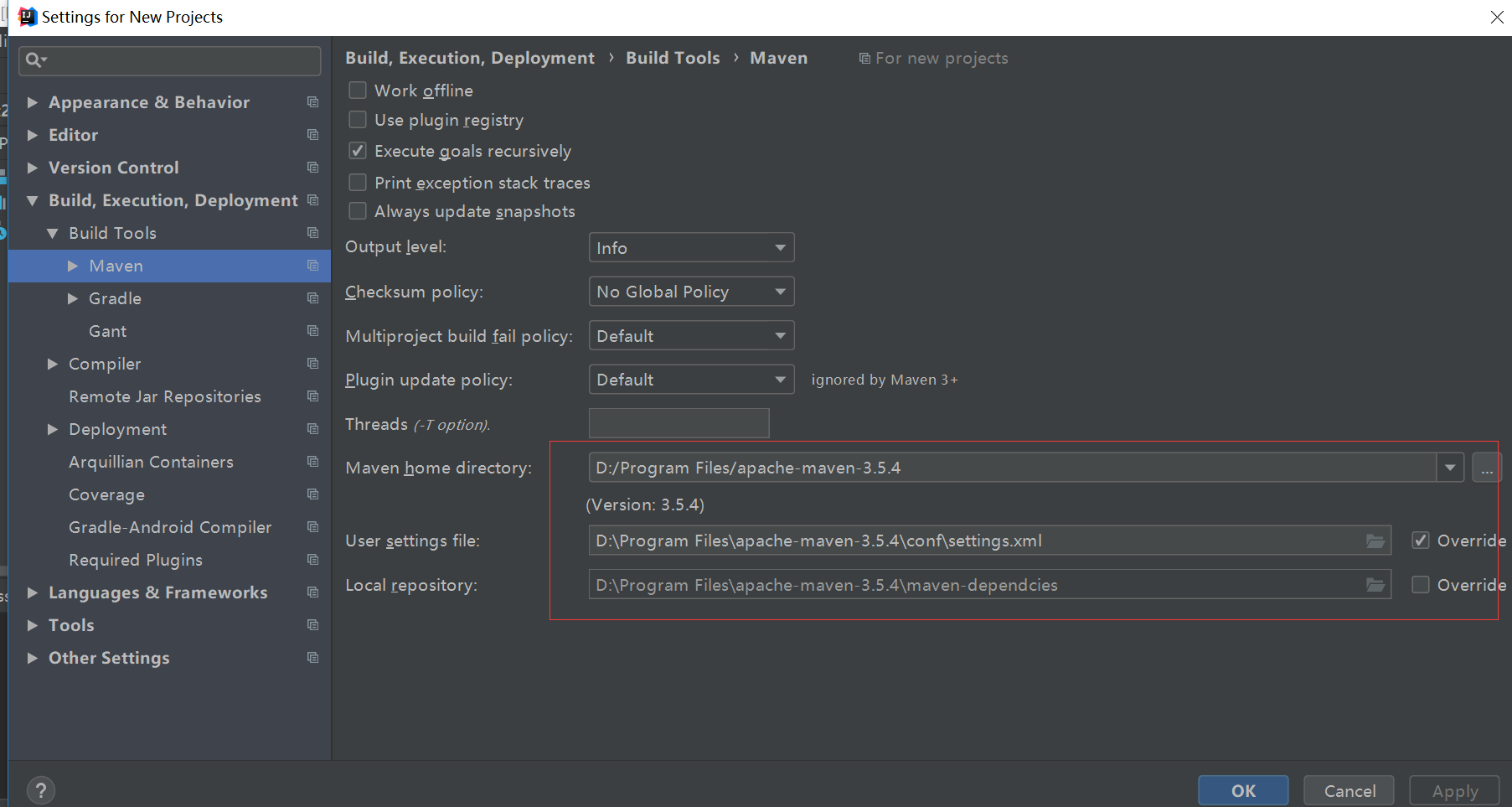
注意要先测试maven是否已经安装成功，配置好本地仓库及下载镜像

上述配置目前没有配置成全局配置，新建项目时会需要重新配置maven，很繁琐如下跟我前面的配置版本不一致：

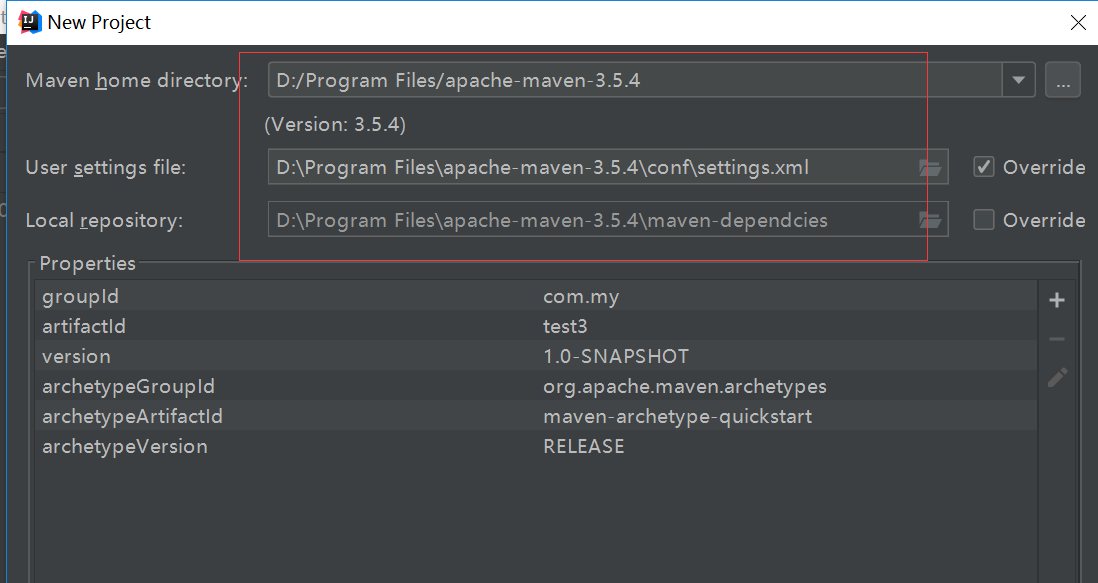


配置maven全局配置：



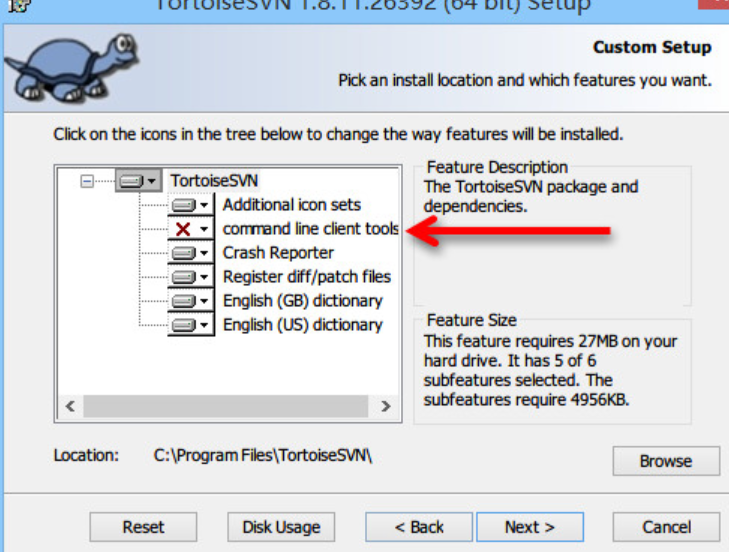


配置后，新建一个maven项目就无需更改maven了。

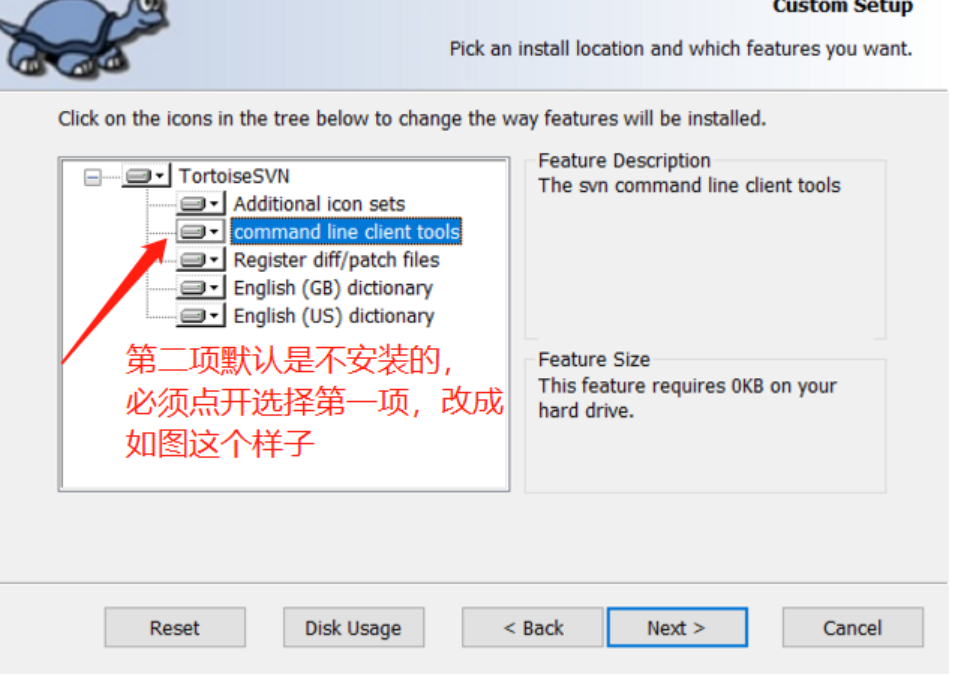


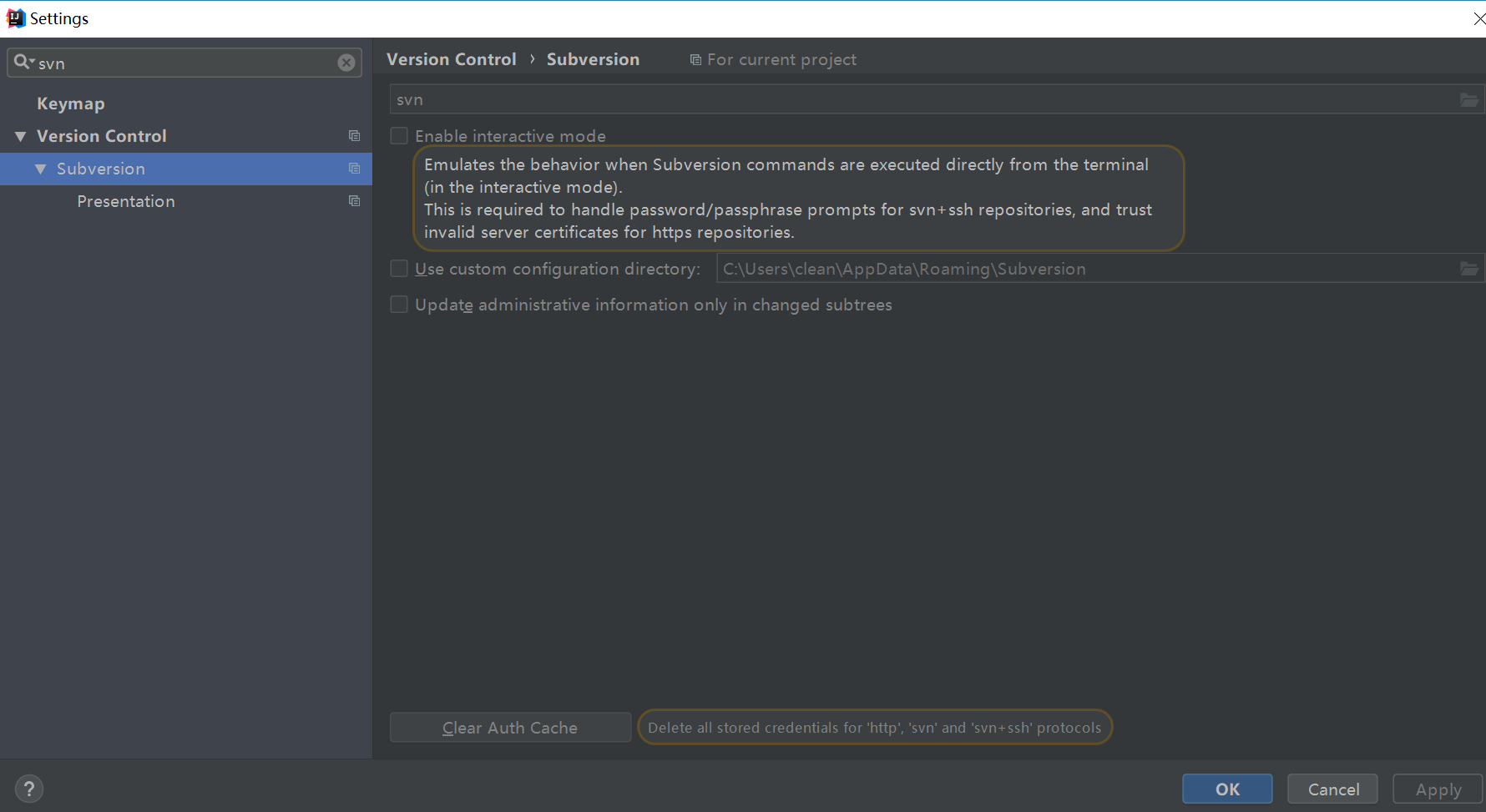
1. Svn配置

前提必须要安装好tortoiseSVN，并且要安装命令行模式。

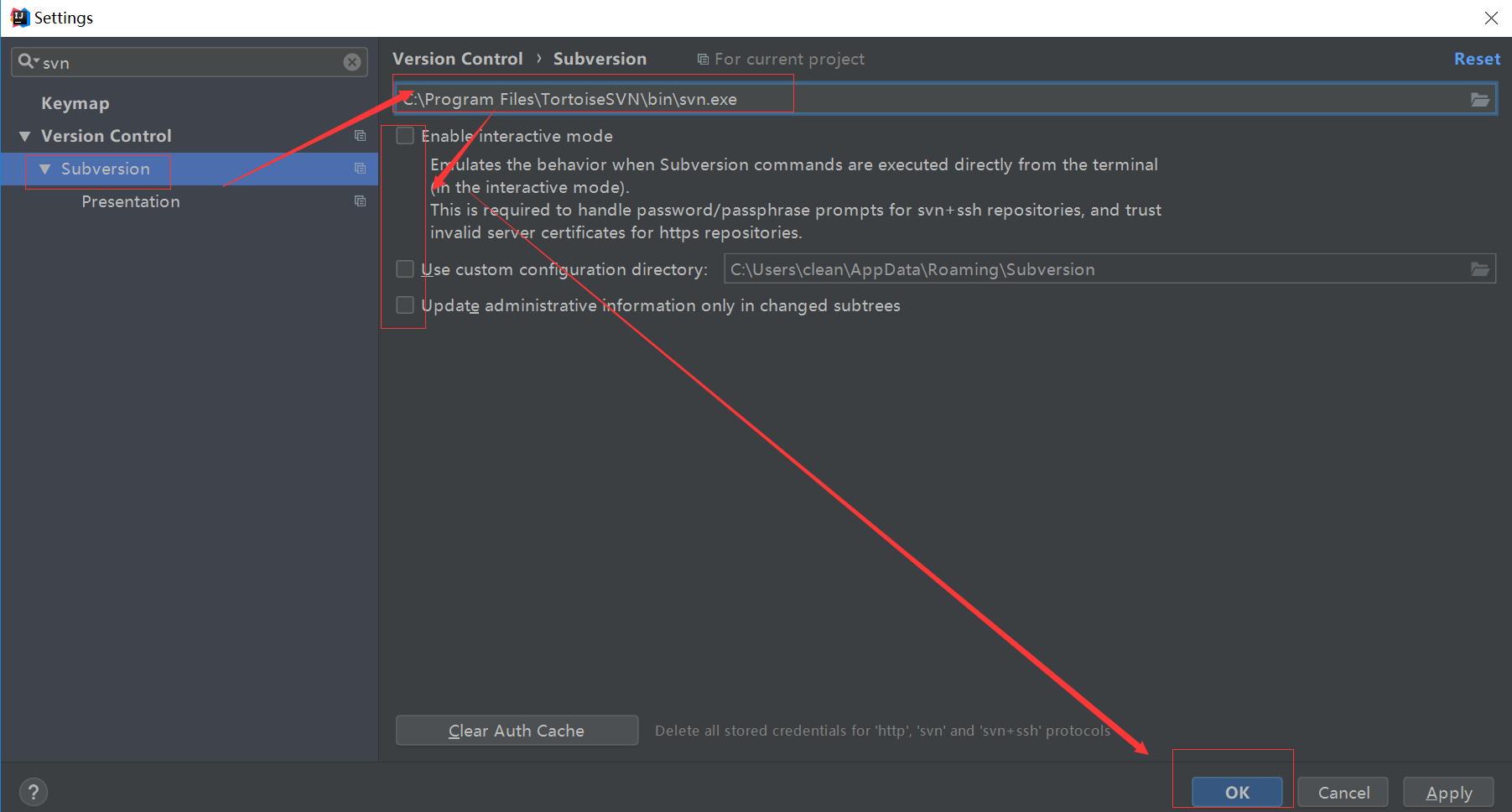


Command line必须要安装。在安装 TortoiseSVN 的时候，默认 command line client tools，是不安装的，这里必须勾选上。



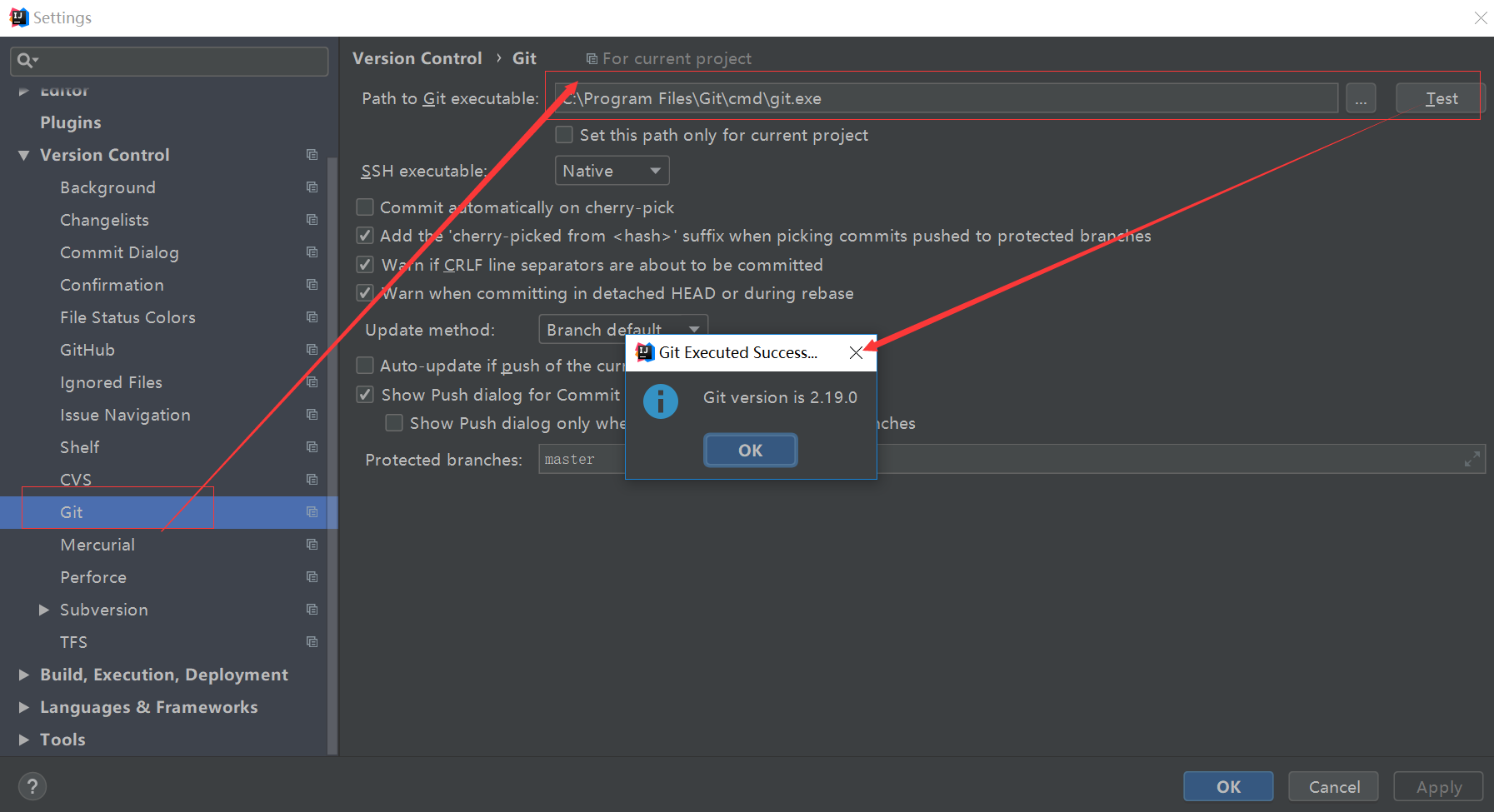


File----->settings-----> Version Control----->Subversion（也可快捷键Crol+Alt+s进入settings）



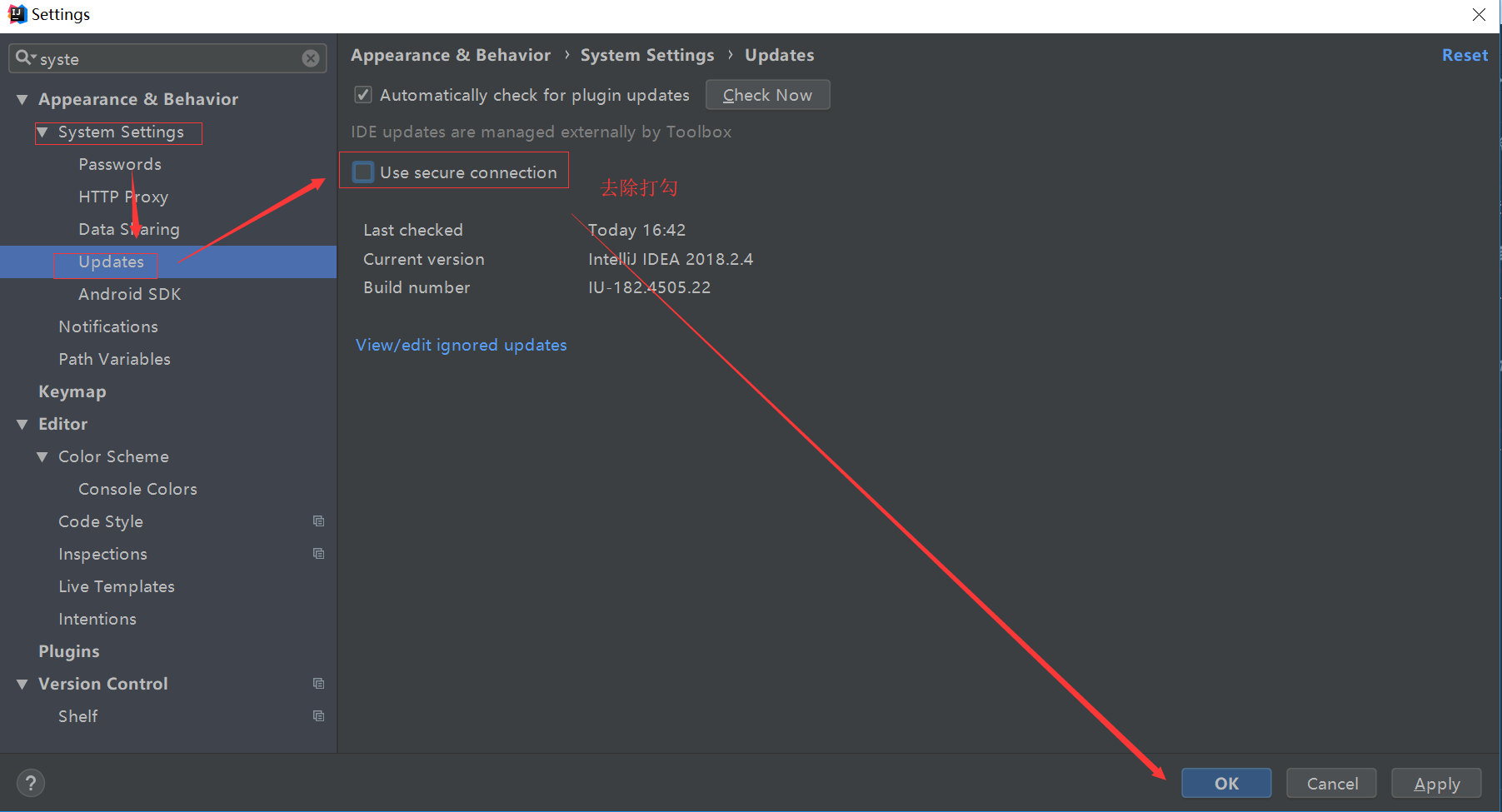
1. Git配置

在File-->Setting->Version Control-->Git-->Path to Git executable选择你的git安装后的git.exe文件，然后点击Test，测试是否设置成功

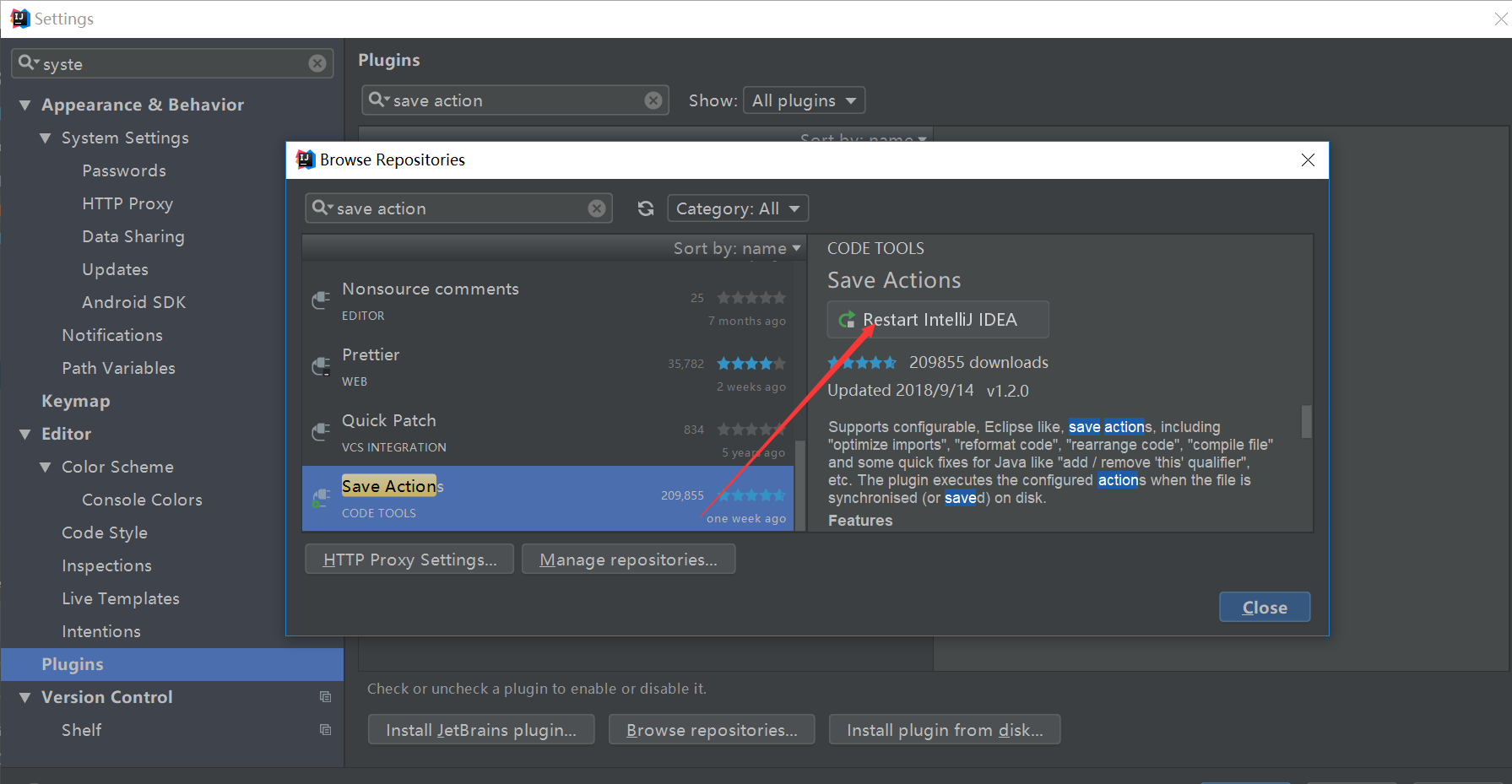


可参考：https://www.cnblogs.com/hkgov/p/7694101.html

1. IDEA无法下载plugin的解决办法
2. system settings中配置



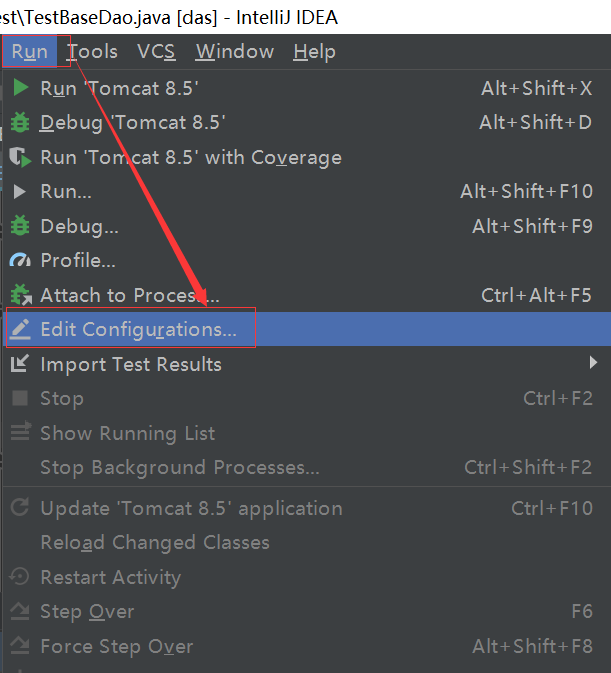
亲测可以：



1. 如果第一种办法还没有解决的话，我们可以从本地磁盘上来安装。

找到settings->Plugins->Install plugin from disk，然后打开IDEA的官网：    <http://plugins.jetbrains.com/> ，打开以后输入想要安装的插件名字，网站会为你罗列出你所搜索的插件列表，然后选择你要安装的版本进行下载，注意这里有很多版本，选择一个跟你IDEA版本匹配的版型，否则会出现版本不兼容无法安装的问题，最好把下载好的plugins文件放在\IntelliJ IDEA 2018.2.4\plugins文件夹下，和其他插件文件放在一起，下载的压缩包不要解压，然后用Install plugin from disk选择下载好的压缩包进行安装即可。

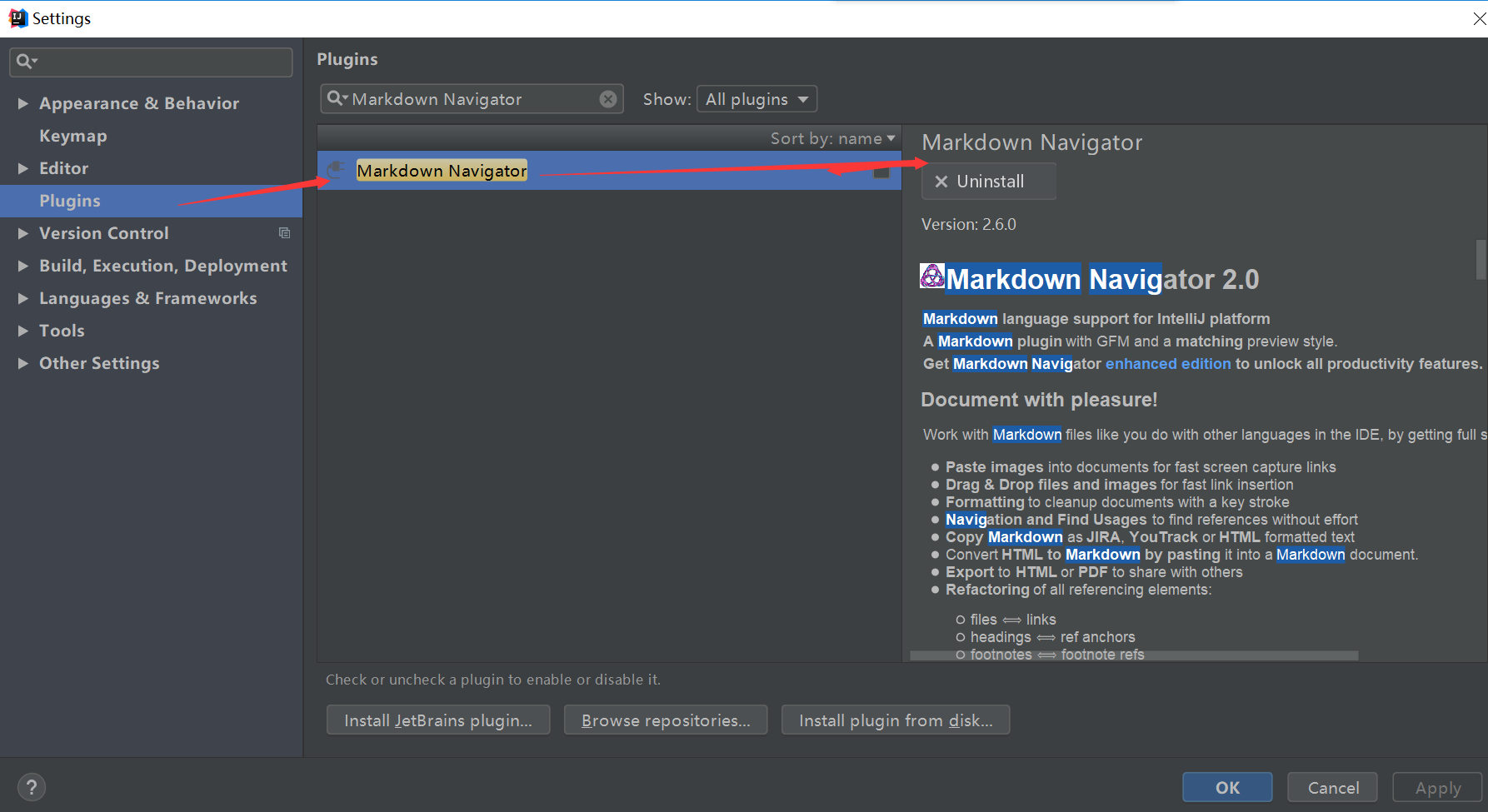
1. 推荐配置
2. 打开Run Dashboard



1. Markdown Navigator

在IDEA中查看和编辑Markdown文件的插件, 功能强大, 需要破解

[开源破解方案] <https://github.com/CrazyBunQnQ/multimarkdown>

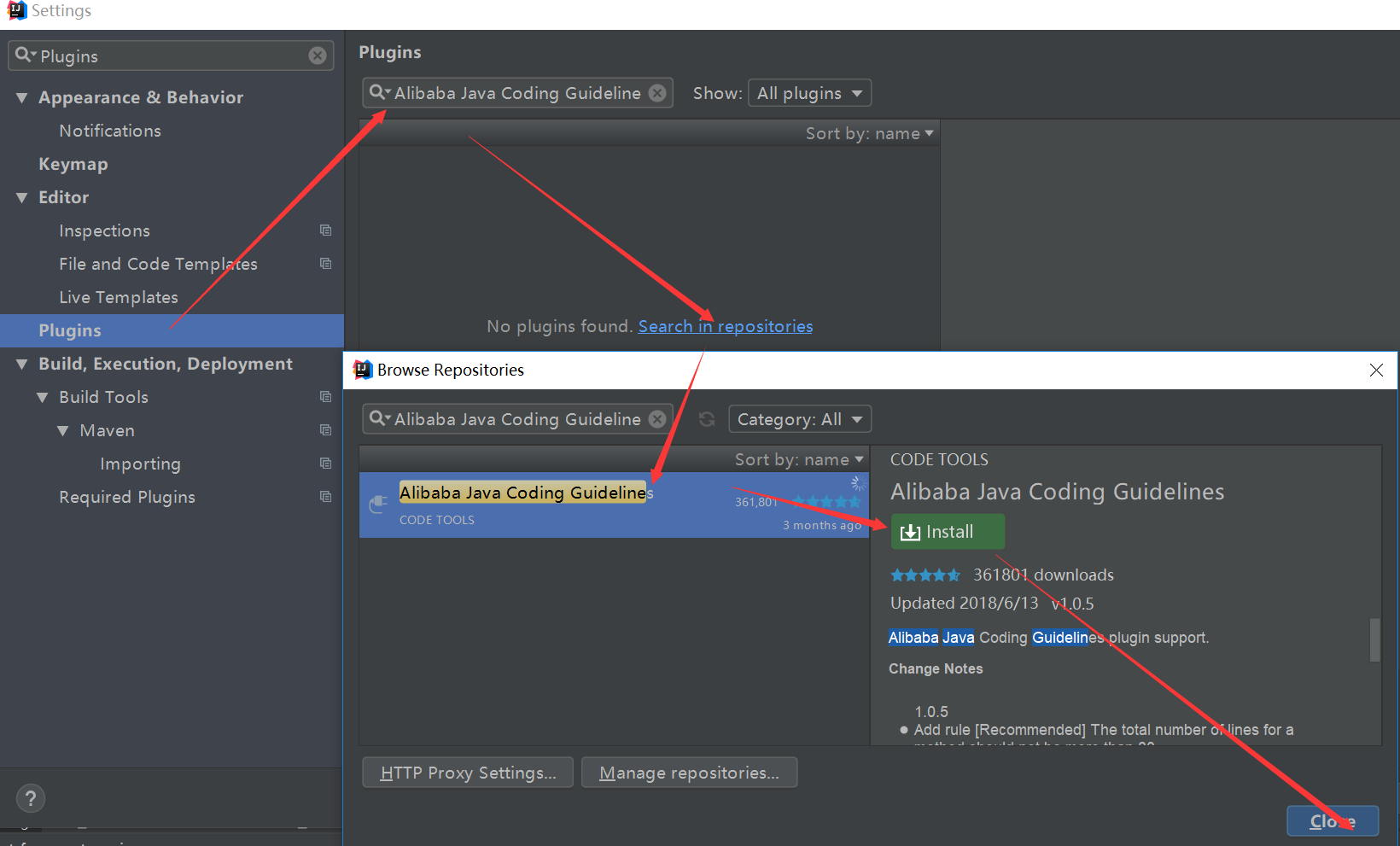


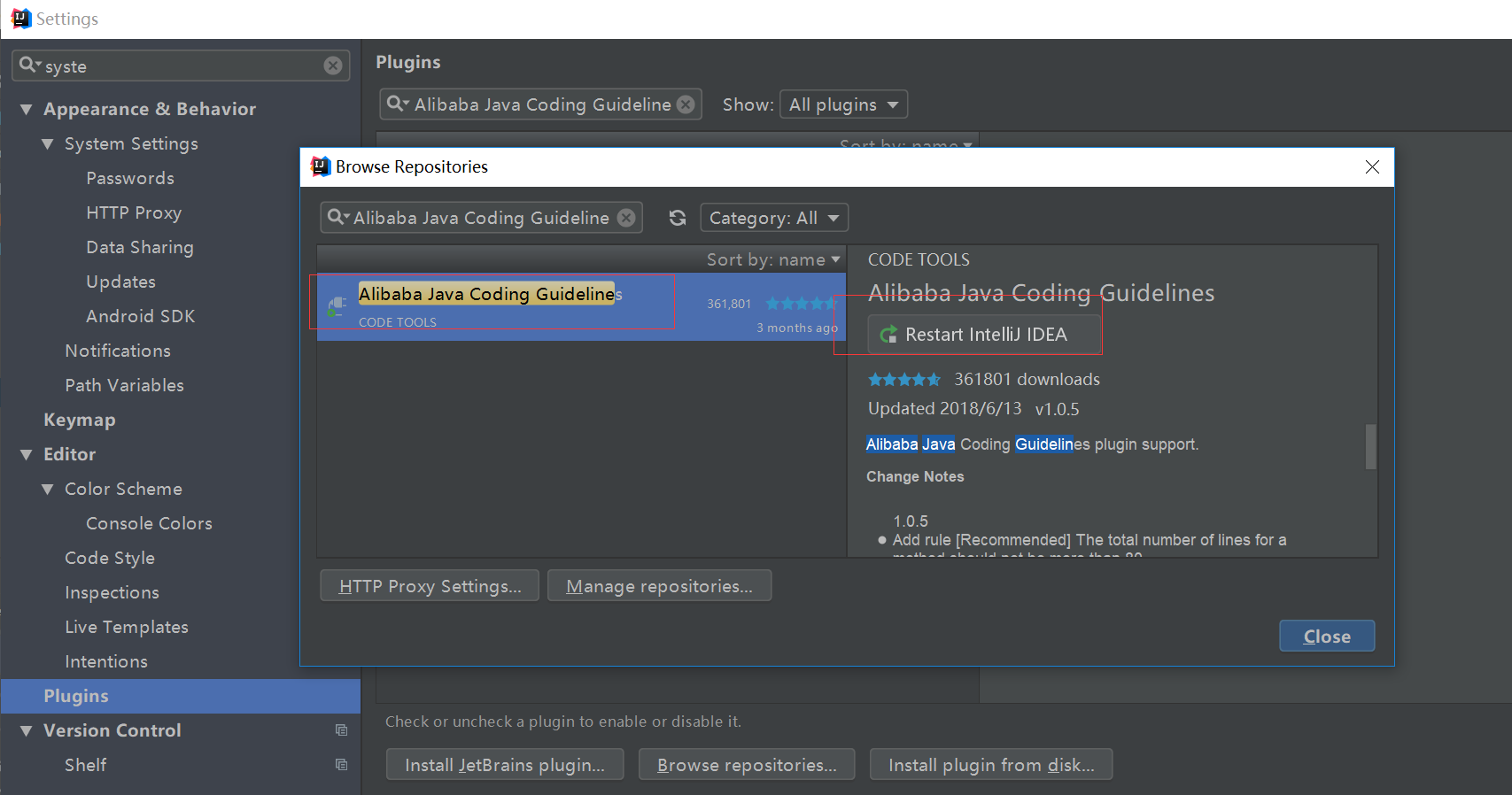
添加文件类型：File → Editor → FileTypes →在Recognized File Type中选择Markdown Language →在Registered Patterns中添加文件类型\*.md;

1. 必装插件
2. Alibaba Java Coding Guideline

阿里公司出品的Java编程手册的检查插件, 当你的编码不满足其规范时将收

到警告。

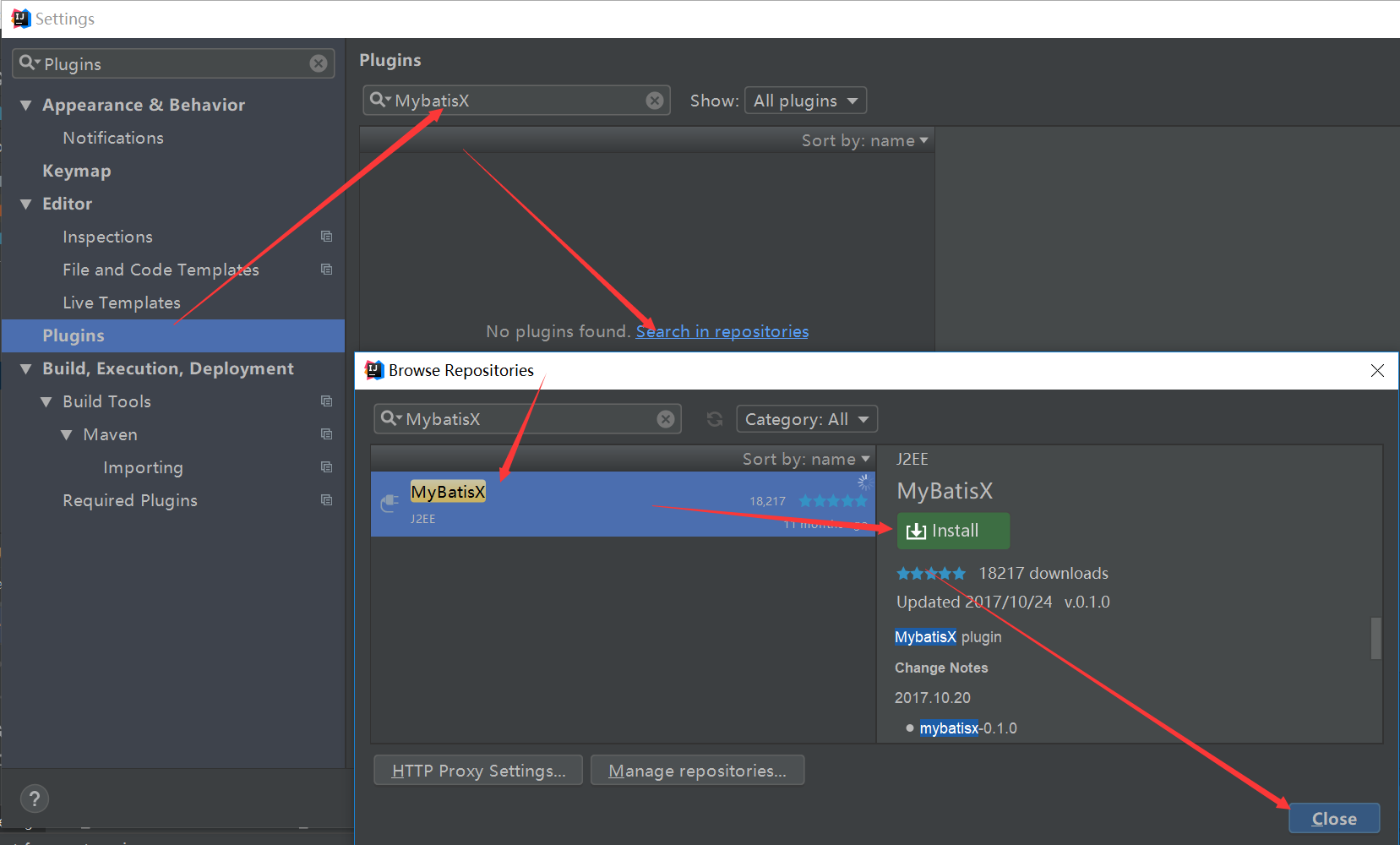


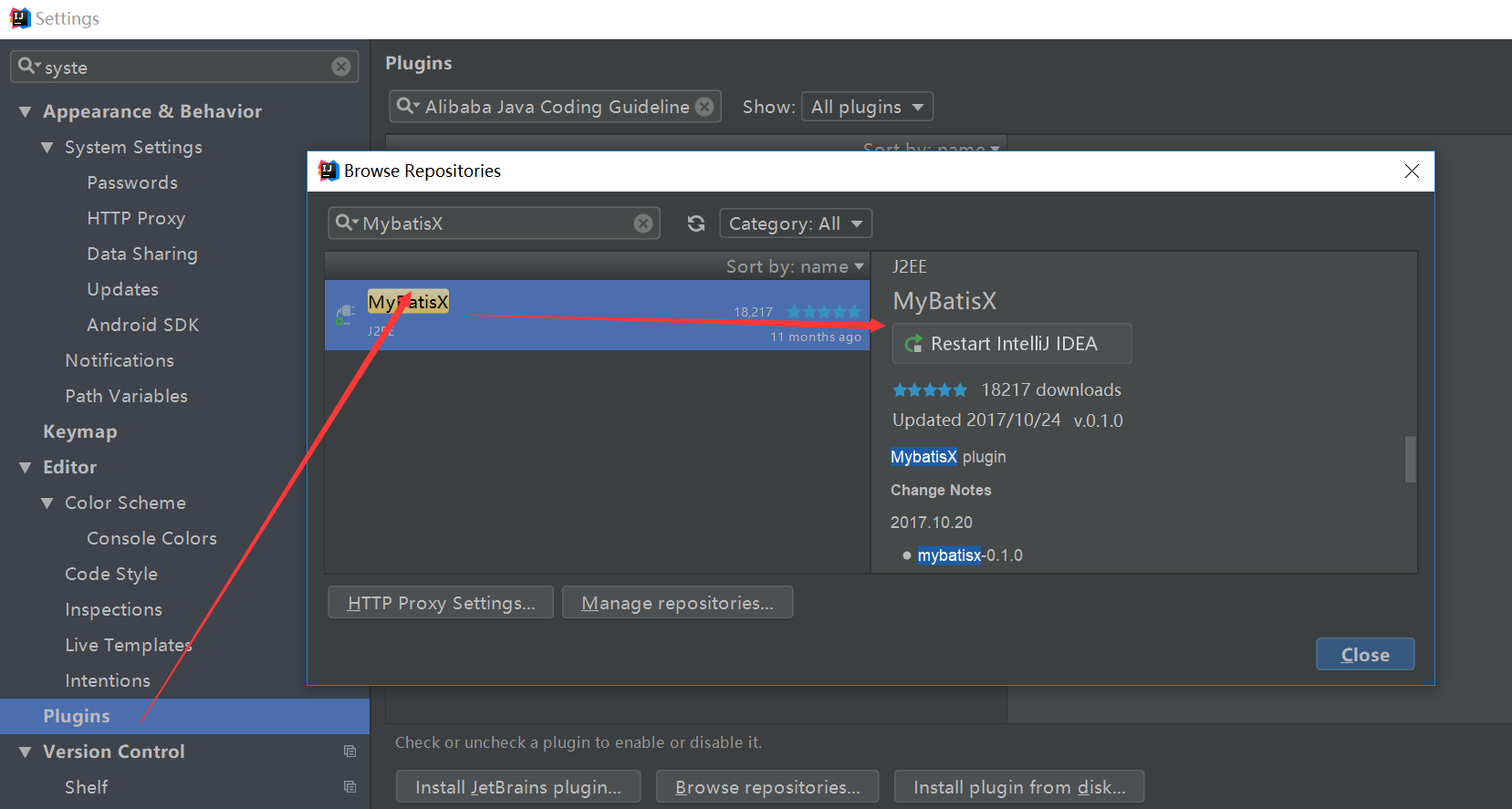


1. MybatisX

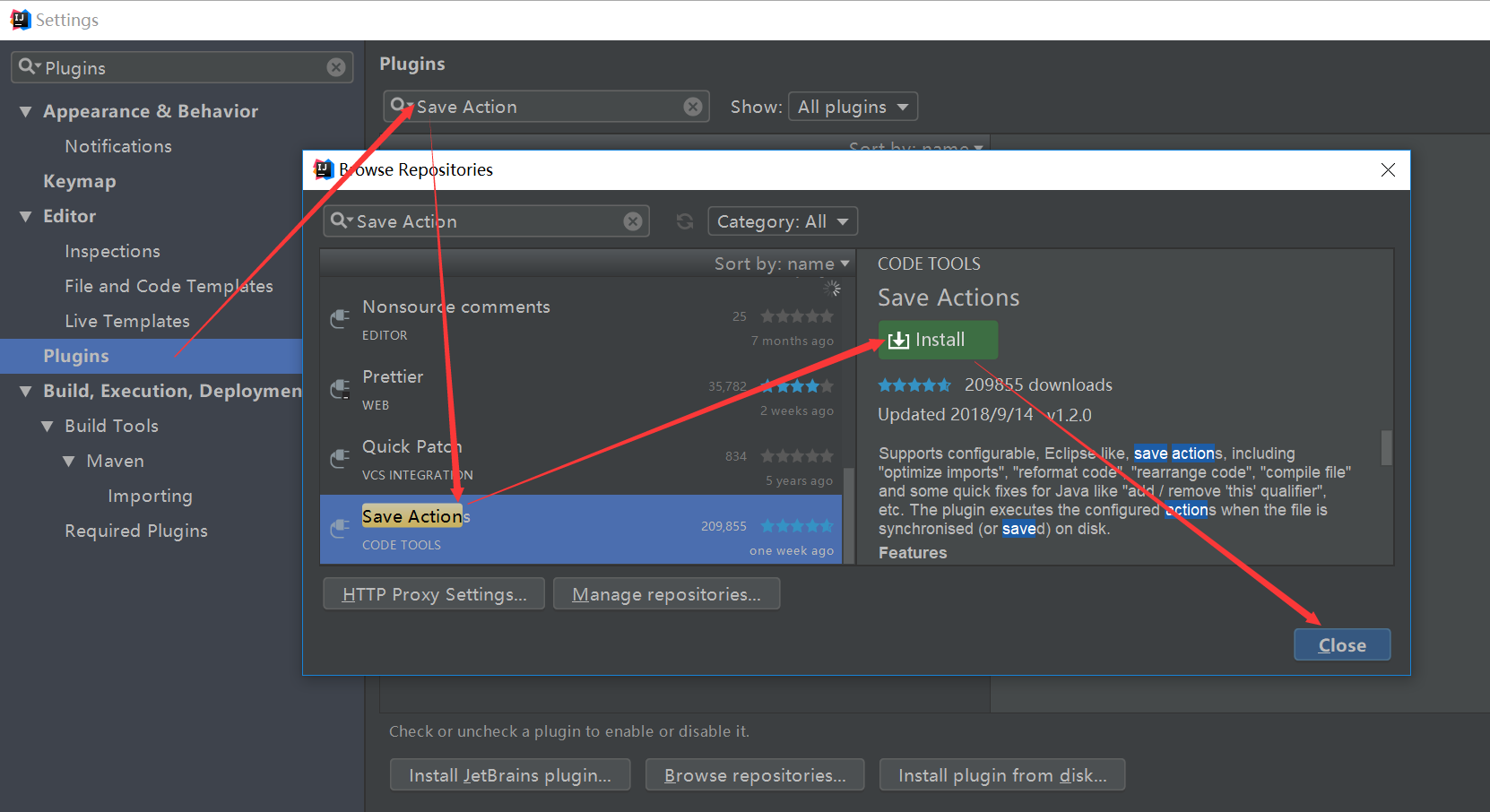
Mybatis-Plus出品的快速开发插件, 能够实现Mapper类和XML文件的互相跳

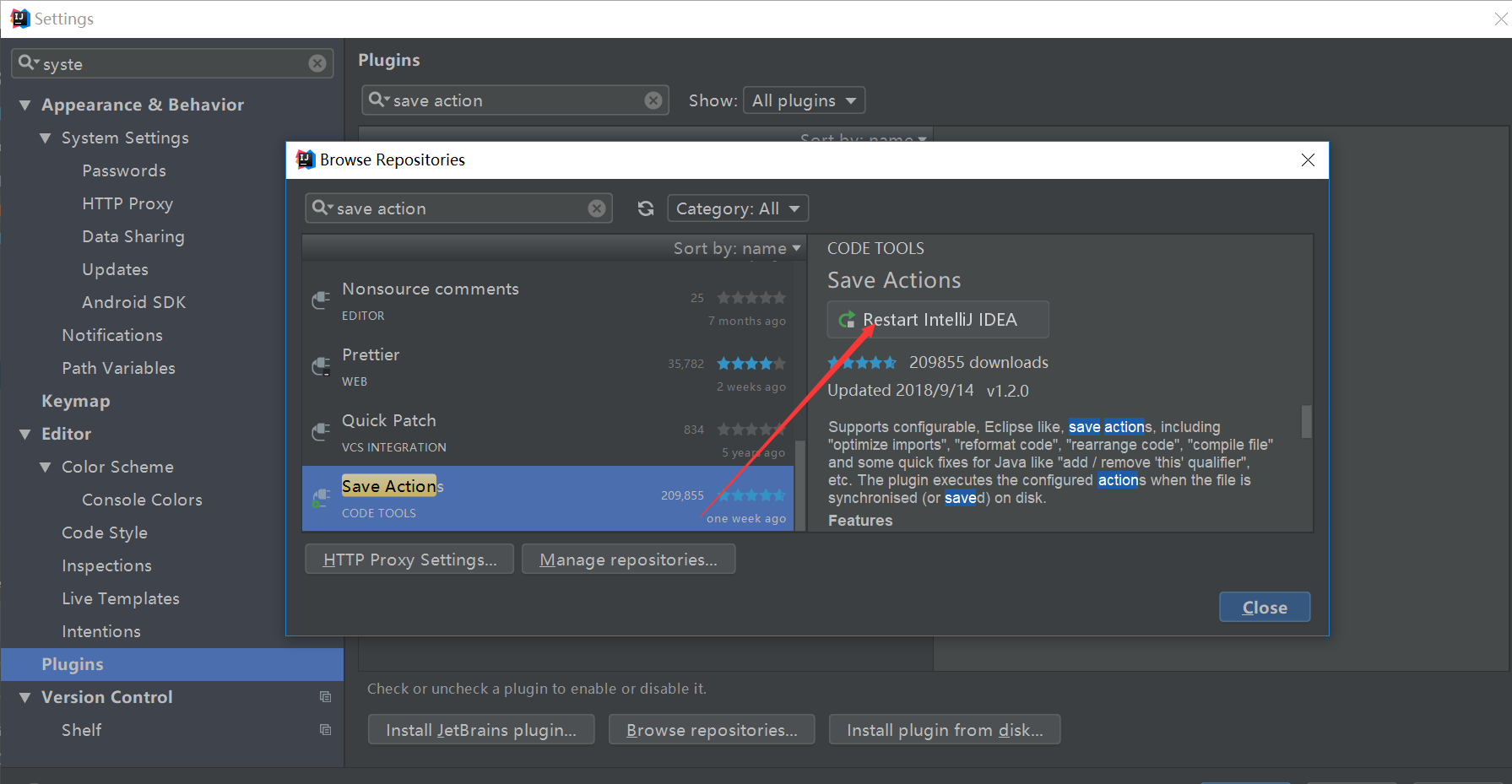
转等操作





1. Save action

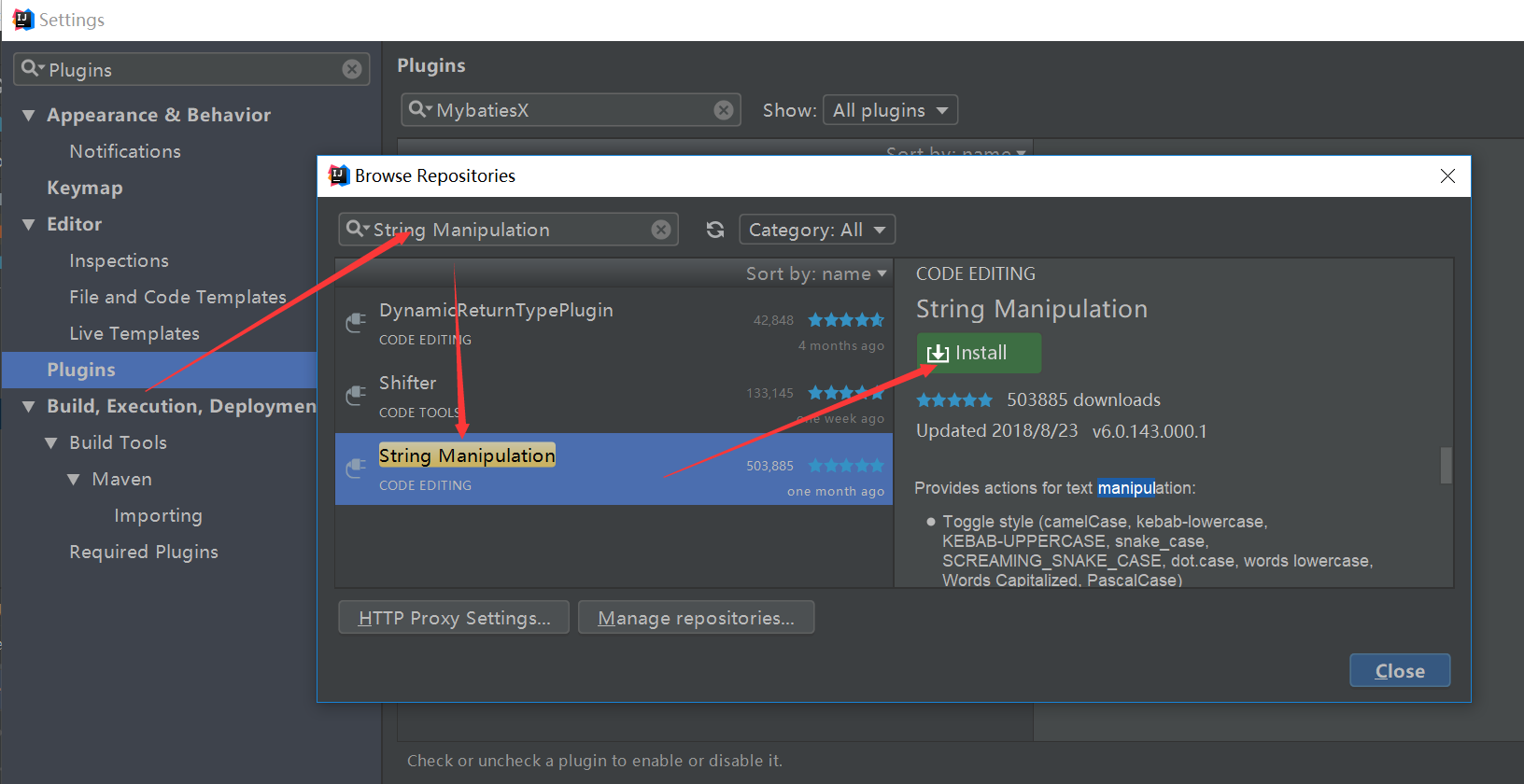


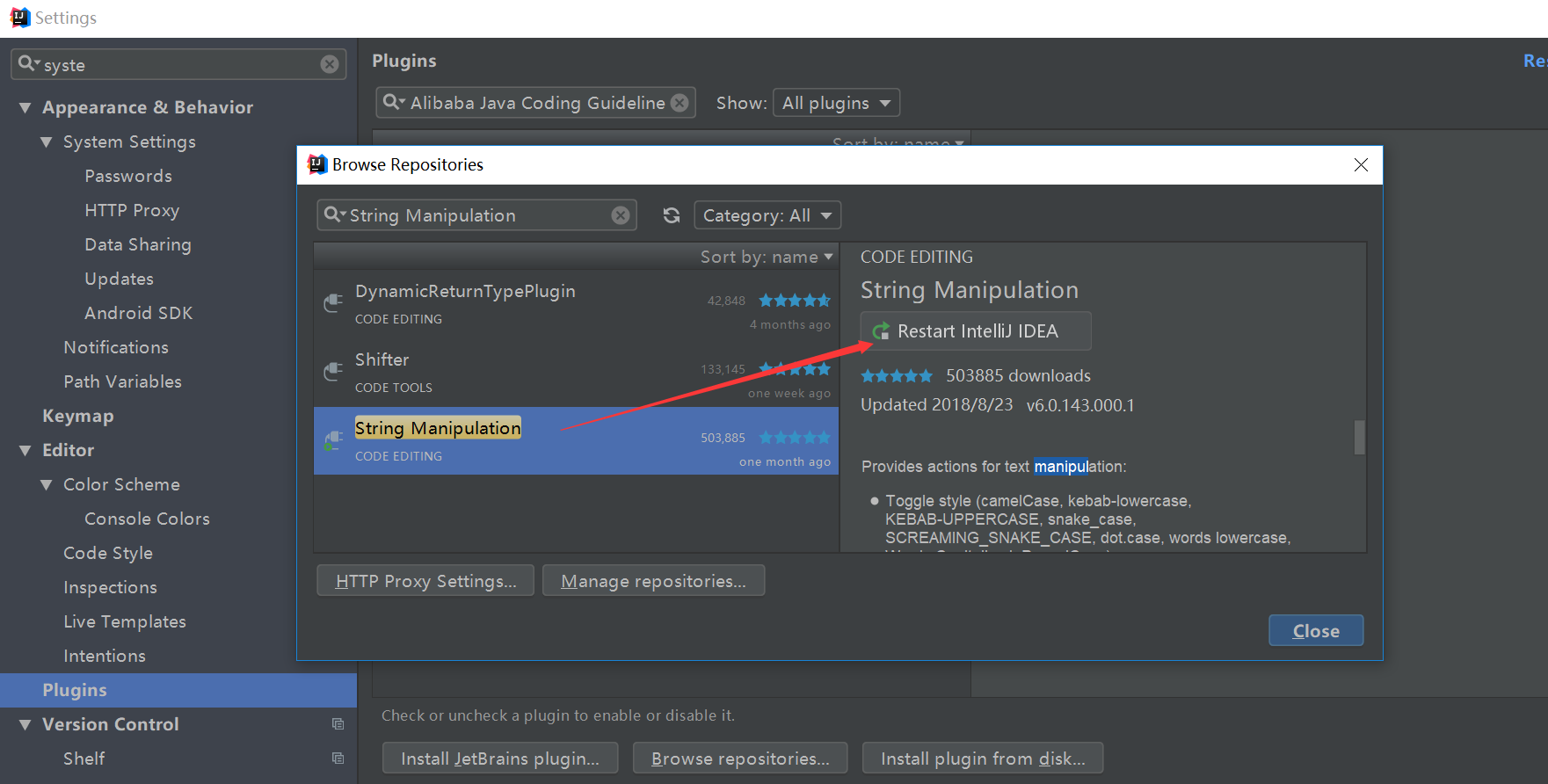


1. 推荐插件
2. String Manipulation

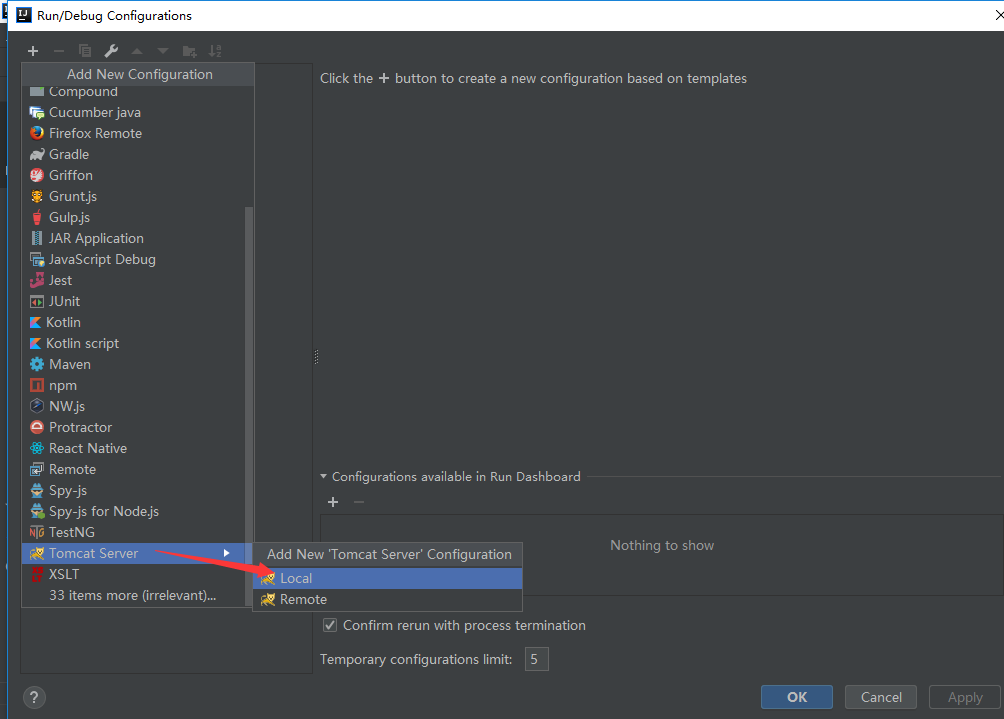
字符串风格转化的插件, 按Alt+M启动, 可以实现驼峰和下划线风格互转, 支持

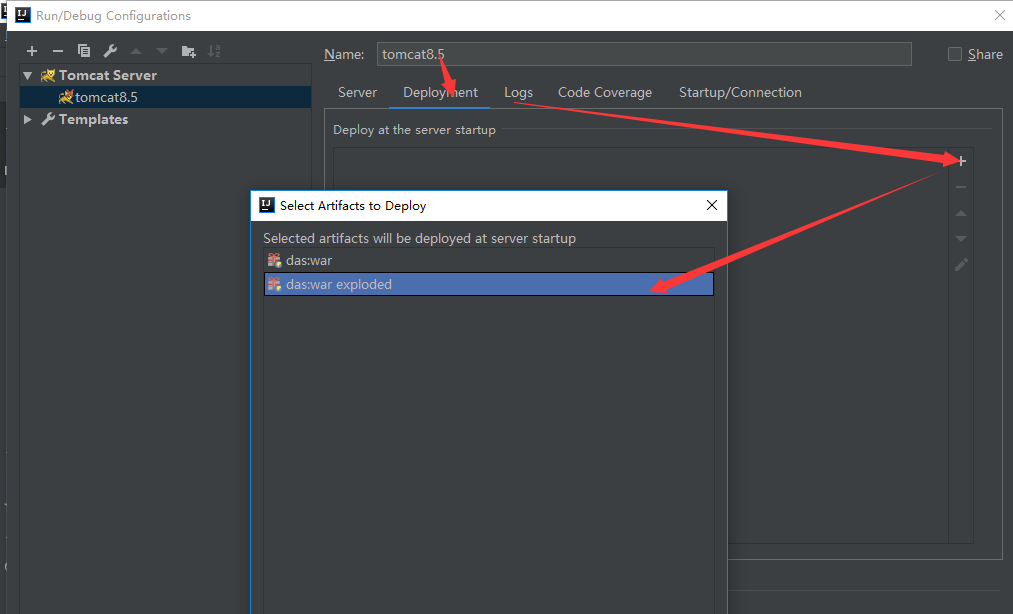
多种风格。

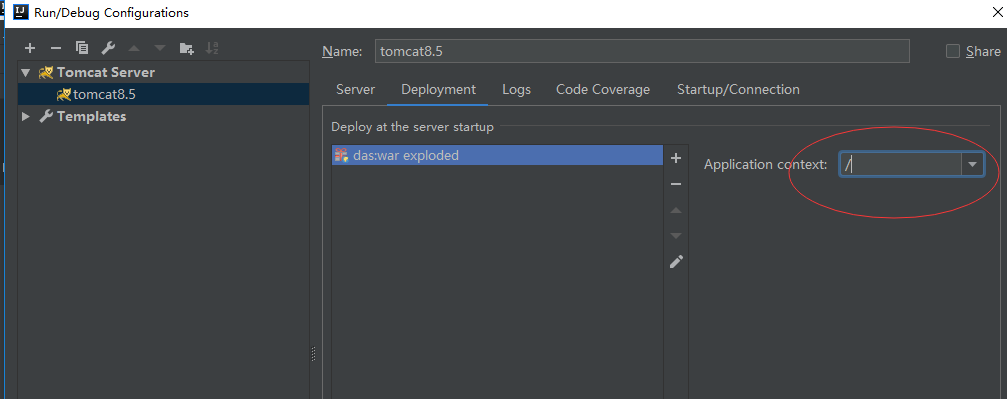


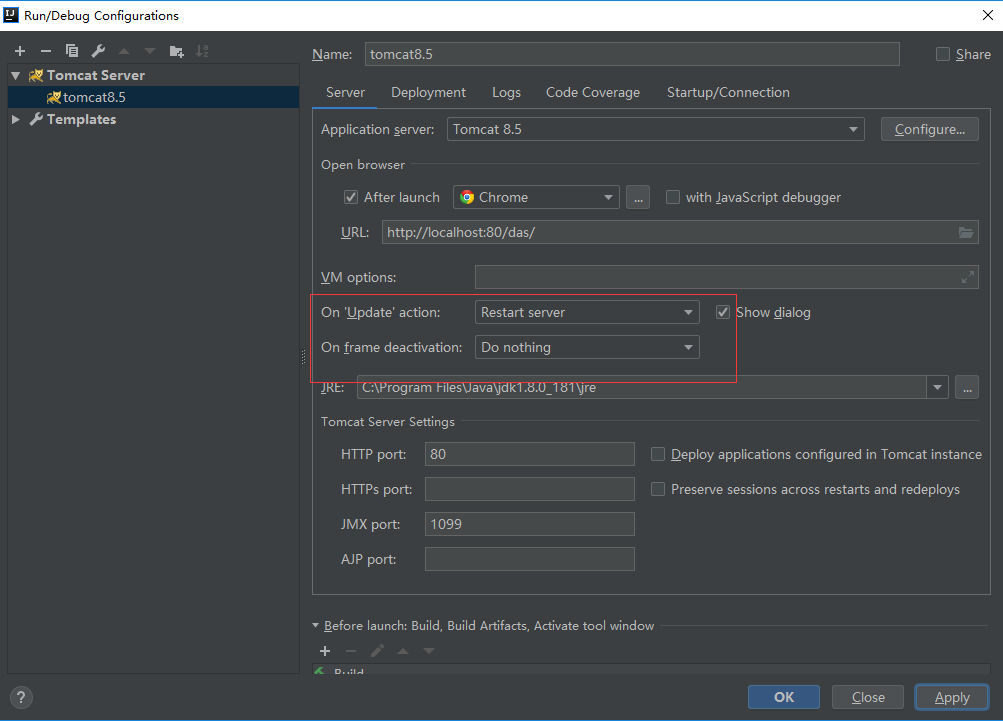


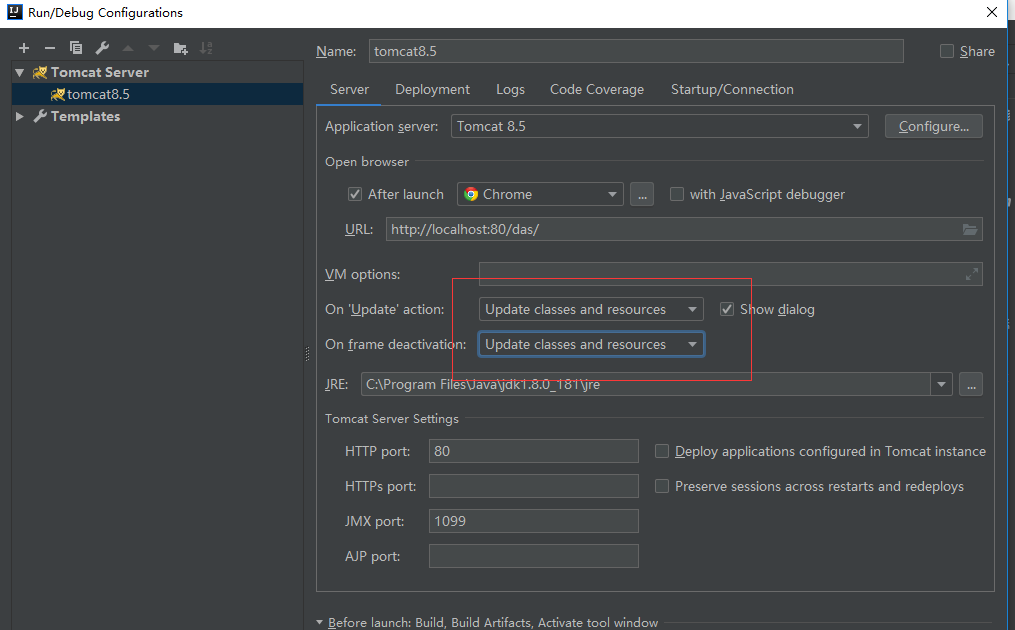
1. Idea导入eclipse项目并部署运行











1. Idea创建maven jar、war、pom项目

https://blog.csdn.net/myarrow/article/details/50824793

1. IntelliJ IDEA常用快捷键汇总

 在使用IntelliJ Idea的时候，使用快捷键是必不可少的。掌握一些常用的快捷键能大大提高我们的开发效率。有些快捷键可以熟练的使用，但是还有另外一些快捷键虽然很好用，但是由于因为没有形成使用习惯或者没有理解快捷键的用法，甚至之前对一些快捷键根本没有概念，导致不会去使用。对于这些快捷键，如果能够用好，编辑代码的效率必能提高一个水平。所以在此梳理出来，加强自己的使用，形成习惯。

*（注：有些操作的快捷键做了更改，和IntelliJ Idea默认的快捷键不一样）*

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
|  |  |  |
| Move Caret to Code Block End | Ctrl+] | 诸如{}围起来的代码块，使用该快捷键可以快速跳转至代码块的结尾处 |
| Move Caret to Code Block Start | Ctrl+[ | 同上，快速跳至代码块的开始出 |
| Complete Current Statement | Ctrl+Shift+Enter | 将输入的if、for、函数等等补上{}或者;使代码语句完整 |
| Start New Line | Shift+Enter | 在当前行的下方开始新行 |
| Start New Line Before Current | Ctrl+Alt+Enter | 在当前行上方插入新行 |
| Delete to Word End | Ctrl+Delete | 删除光标所在至单词结尾处的所有字符 |
| Delete to Word Start | Ctrl+BackSpace | 删除光标所在至单词开头的所有字符 |
| Move Caret to Previous Word | Ctrl+向左箭头 | 将光标移至前一个单词 |
| Move Caret to Next Word | Ctrl+向右箭头 | 将光标移至后一个单词 |
| Scroll Up | Ctrl+向上箭头 | 向上滚动一行 |
| Scroll Down | Ctrl+向下箭头 | 向下滚动一行 |
| Extend Selection | Ctrl+W | 选中整个单词 |
| Toggle Case | Ctrl+Shift+U | 切换大小写 |

———————————————Edit——————————————————–

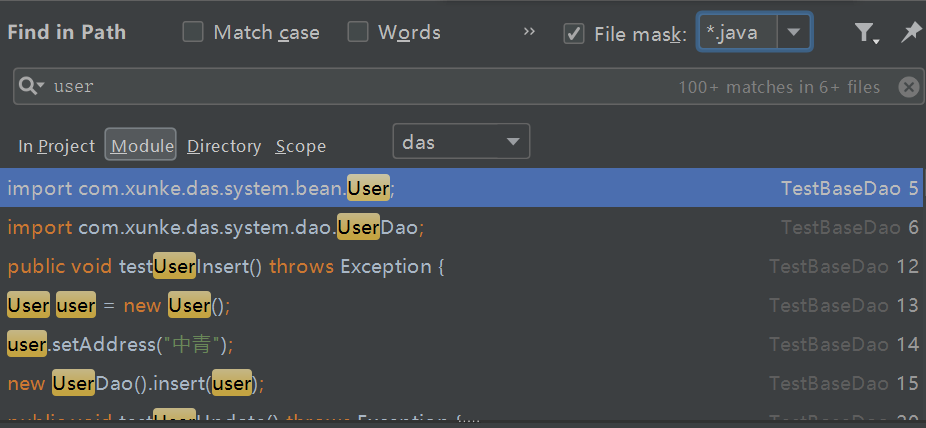
| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Undo | Ctrl+Z | 撤销 |
| Redo | Ctrl+Shift+Z | 重做 |
| Cut | Ctrl+X | 剪切 |
| Copy | Ctrl+C | 复制 |
| Paste | Ctrl+V | 粘贴 |
| Join Lines | Ctrl+Shift+J | 将选中的行合并成一行 |

—————-Find—————–

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Find | Ctrl+F | 在当前文件中查找 |
|  | Ctrl+H | 在全局文件中查找字符串 |
| Find in Path | Ctrl+Shift+F | 在全局文件中查找字符串 |
| Replace in Path | Ctrl+Shift+R | 在全局中替换字符串 |
| Find Usages | Alt+F7 | 查找当前变量的使用，并列表显示 |
| Show Usages | Ctrl+Alt+F7 | 查找当前变量的使用，并直接对话框显示 |
| Find Usages in File | Ctrl+F7 | 在文件中查找符号的使用 |
| Highlight Usages in File | Ctrl+Shift+F7 | 在文件中高亮显示变量的使用 |

*这里的快捷键用的频率还是很高的，但是之前用的最多的是Ctrl+F和Ctrl+Shift+F，后面相关的Find Usages基本上没有用过，后面应该多使用，有的时候相对Ctrl+F在文件内按字符串查找，还是更好用一些*

Ctrl+H



—————————————–Navigate————————————————

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Class… | Ctrl+N | 查找类文件 |
| File… | Ctrl+Shift+N | 查找文件 |
| Line… | Ctrl+G | 定位到文件某一行 |
| Back | Alt+向左箭头 | 返回至上次光标位置 |
| Forward | Alt+向右箭头 | 返回至后一次光标位置 |
| Last Edit Location | Ctrl+Shift+Backspace | 返回上次编辑位置 |
| Next Edit Location | Ctrl+Shift+反斜杠 | 返回后一次编辑位置 |
| Declaration | Ctrl+B | 定位至变量定义的位置 |
| Implementation(s) | Ctrl+Alt+B | 定位至选中类或方法的具体实现 |
| Type Declaration | Ctrl+Shift+B | 直接定位至光标所在变量的类型定义 |
| Super Method | Ctrl+U | 直接定位至当前方法override或者implements的方法定义处 |
| File Structure | Ctrl+F12 | 显示当前文件的文件结构 |
| File Path | Ctrl+Alt+F12 | 显示当前文件的路径，并可以方便的将相关父路径打开 |
| Type Hierarchy | Ctrl+H | 显示当前类的继承层次 |
| Method Hierarchy | Ctrl+Shift+H | 显示当前方法的继承层次 |
| Call Hierarchy | Ctrl+Alt+H | 显示当前方法的调用层次 |
| Next Highlighted Error | F2 | 定位至下一个错误处 |
| Previous Highlighted Error | Shift+F2 | 定位至前一个错误处 |
| Previous Occurrence | Ctrl+Alt+向上箭头 | 查找前一个变量共现的地方 |
| Next Occurrence | Ctrl+Alt+向下箭头 | 查找下一个变量共现的地方 |

*目前还不知道Previous Occurrence 和 Next Occurrence是怎么用的，在变量上使用没有反应。不过在Edit–Find菜单下有几个菜单项：Find Next \/ Move to Next Occurrence、Find Previous \/ Move to Previous Occurrence等。当选中变量的时候，需要首先点击“Find Word at Caret”，然后再点击上述选项才有用*

————————————————-Code———————————————–

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Override Methods… | Ctrl+O | 重写基类的方法 |
| Implement Methods… | Ctrl+I | 实现基类或接口中的方法 |
| Generate… | Alt+Insert | 产生构造方法、getter/setter等方法 |
| Surround With… | Ctrl+Alt+T | 将选中的代码使用if、while、try/catch等包装 |
| Unwrap/Remove… | Ctrl+Shift+Delete | 去除相关的包装代码 |

—————————————–Completion——————————————

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Basic | Alt+/ | 自动完成 |
| SmartType | Alt+Enter | 自动提示完成 |

—————————————-Folding————————————————-

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Expand | Ctrl+= | 展开代码 |
| Collapse | Ctrl+- | 收缩代码 |
| Expand Recursively | Ctrl+Alt+= | 递归展开代码 |
| Collapse Recursively | Ctrl+Alt+- | 递归收缩代码 |
| Expand All | Ctrl+Shift+= | 展开所有代码 |
| Collapse All | Ctrl+Shift+- | 收缩所有代码 |

———————————

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Insert Live Template | Ctrl+J | 插入Live Template |
| Surround with Live Template | Ctrl+Alt+J | 使用Live Template包装 |
| Comment with Line Comment | Ctrl+/ | 使用//进行注释 |
| Comment with Block Comment | Ctrl+Shift+/ | 使用/\*\*/进行注释 |
| Reformat Code | Ctrl+Alt+L | 格式化代码 |
| Auto-Indent Lines | Ctrl+Alt+I | 自动缩进行 |
| Optimize Imports | Ctrl+Alt+O | 优化import |

———————————

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Move Statement Down | Ctrl+Shift+向下箭头 | 将光标所在的代码块向下整体移动 |
| Move Statement Up | Ctrl+Shift+向上箭头 | 将光标所在的代码块向上移动 |
| Move Element Left | Ctrl+Alt+Shift+向左箭头 | 将元素向左移动 |
| Move Element Right | Ctrl+Alt+Shift+向右箭头 | 将元素向右移动 |
| Move Line Down | Alt +向下箭头 | 将行向下移动 eclipse |
| Move Line Up | Alt +向上箭头 | 将行向上移动 eclipse |

————————————-Refactor——————————————–

| **动作** | **快捷键** | **说明** |
| --- | --- | --- |
| Rename | Shift+F6 | 重命名 |
| Change Signature | Ctrl+F6 | 更改函数签名 |
| Type Migration | Ctrl+Shift+F6 | 更改类型 |