转 IntelliJ IDEA详细配置和使用教程

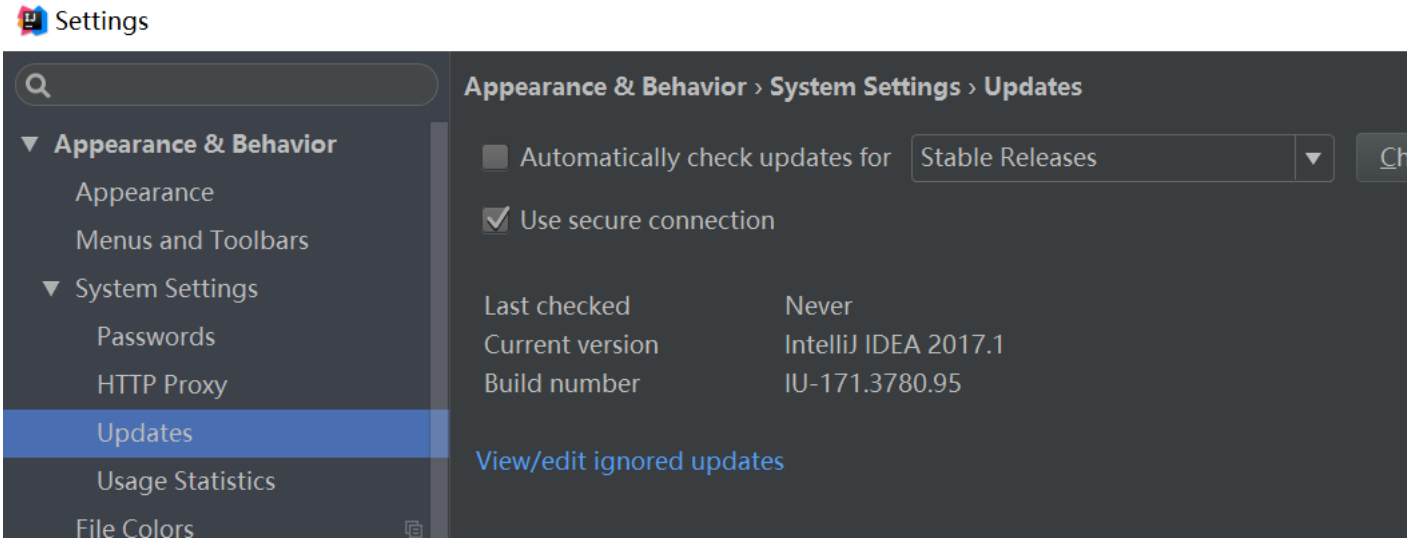
2017年12月26日 11:15:29

前言

正所谓工欲善其事必先利其器，对开发人员而言若想提高编码效率，一款高效的开发工具是必不可少的，相信看到该博客的朋友们都已经对IntelliJ IDE是一名Java开发工程师，所以以下内容均以Java为主。(相信有不少人和我一样是从Eclipse转粉IntelliJ IDEA，在学习IntelliJ IDEA前请尽量忘记Eclipse学习。至于为什么写这篇博文，我的目的是想把自己的个性化配置记录下来，当然如果与此同时能帮助到其他人，岂不美哉，本文将持续更新，由于本

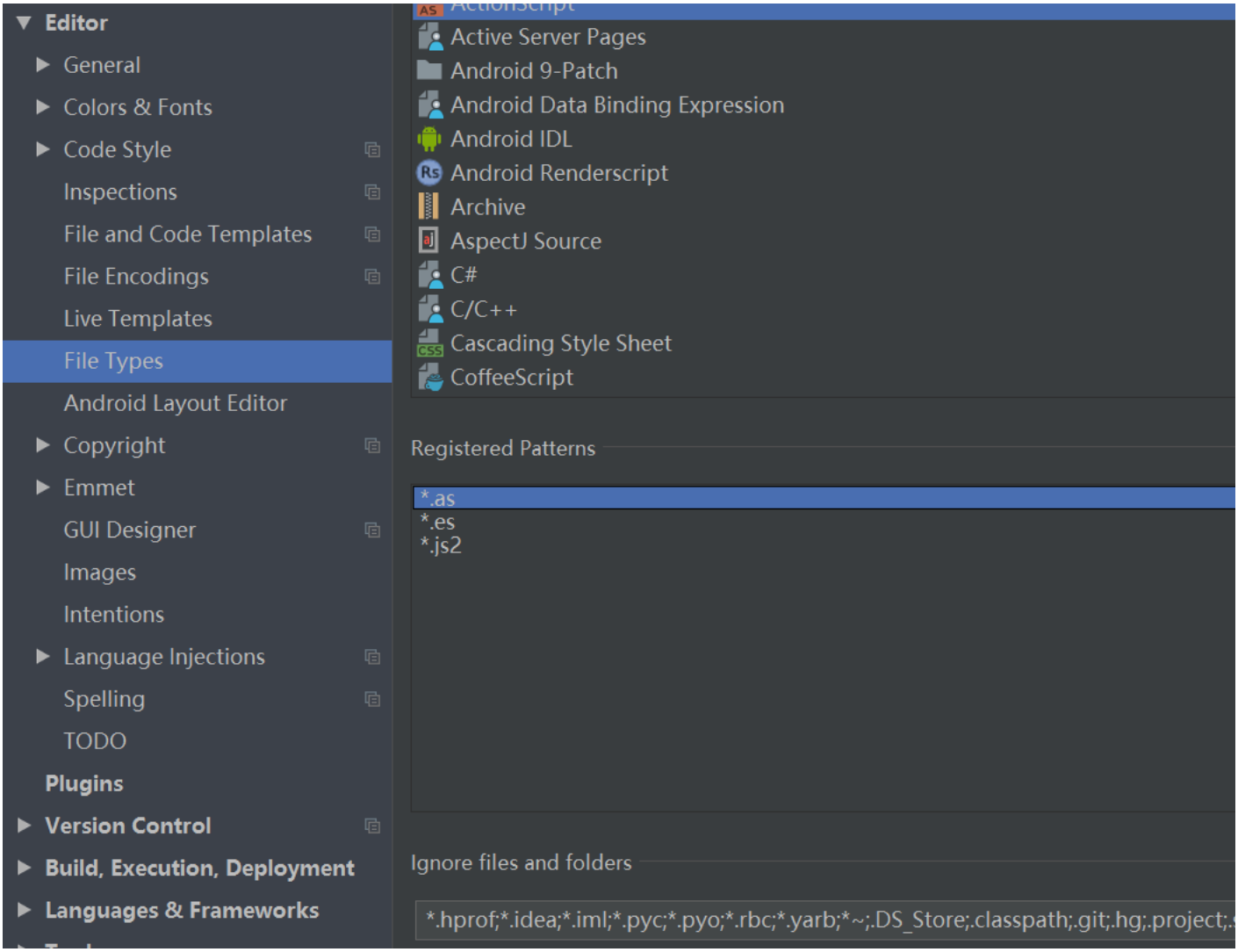
关闭Intellij IDEA自动更新

在File->Settings->Appearance & Behavior->System Settings->Updates下取消Automatically check updates for勾选



隐藏.idea文件夹和.iml等文件

IntelliJ IDEA项目会自动生成一个.idea文件夹和.iml文讲，看着实在是碍眼，所以对以上文件进行隐藏处理
在File->Settings->Editor->File Types下的"Ignore files and folders"一栏添加 *.idea;*.iml;等配置如下图所示



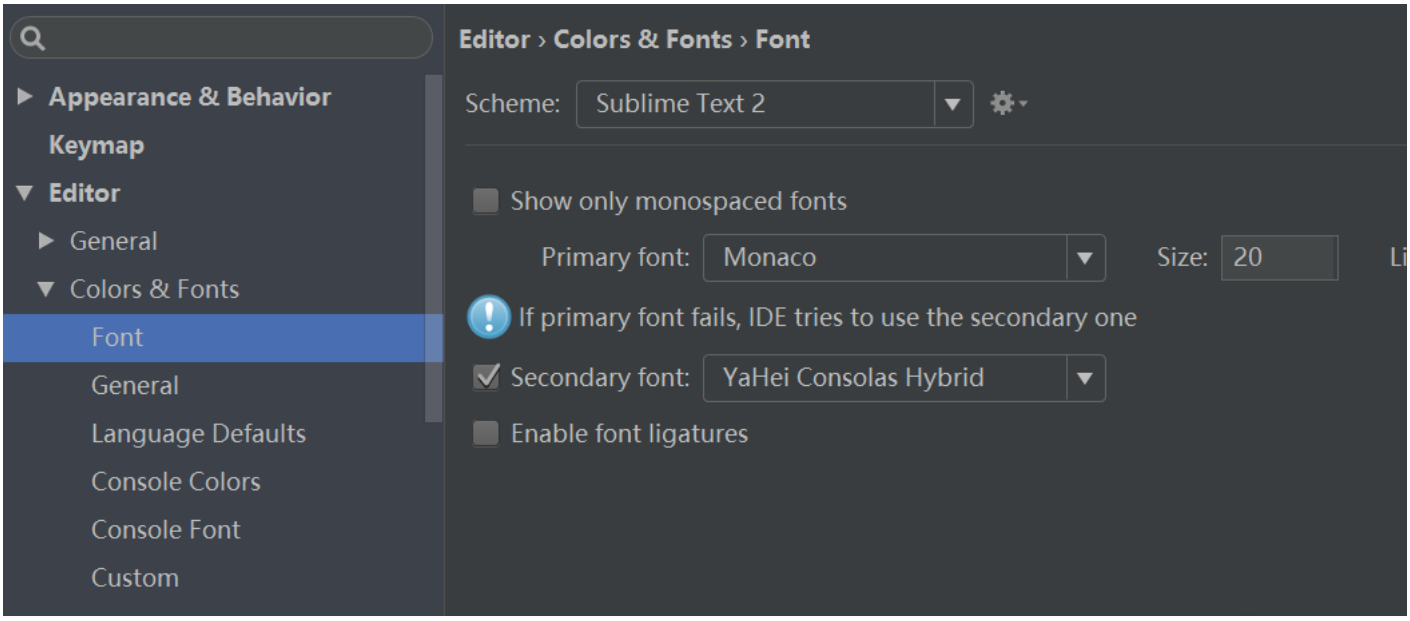
代码编辑器主题风格

编辑器风格修改个人并不推荐完全由自己来配置，因为网上提供了很多优秀的主题风格，我们可以导入自己喜欢的主题，然后在其基础上进行微调，推

```
1.从主菜单打开你的编辑器选择File->Import Setting.选择你下载的Jar文件;
2.等待重启之后进行配置打开File->Settings->Editor->Colors and fonts 然后选择你安装的主题即可完成;
```

```
1
2
```

设置第一字体和第二字体，修改字体大小:
自行去网上下载相应字体安装后重启Intellij IDEA，在主菜单下选择File->Settings->Editor->Colors & Fonts -> Font
show only monospaced fonts表示筛选显示系统上的等宽字体，由于Windows系统上等宽字体并不多，勾选此选项出现的下拉字体可选择就很少，取消
字体使用Yahei Consolas Hybrid，该字体含有中文。字体大小我是修改为20，配置如下图所示:



控制台输出字体和上述类似 此处不进行说明(控制台输出乱码即通过配置字体解决)

文件编码设置

File->Settings->Editor->File Encodings

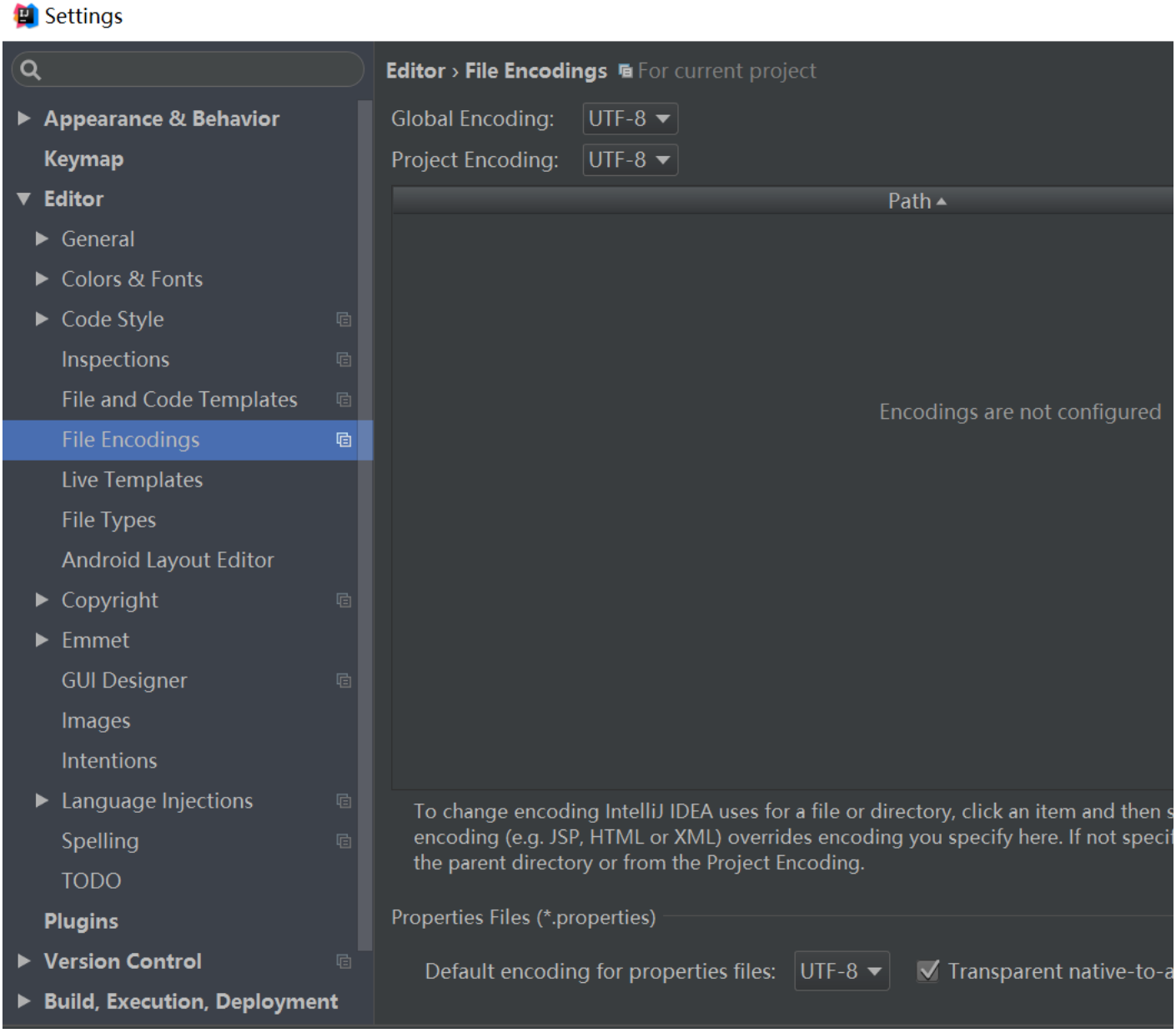
推荐设置

```
Global Encoding:UTF-8
Projectt Encoding:UTF-8
Default encoding for properties files:UTF-8
勾选上Transparent native-to-ascii conversion
```

1
2
3
4

Transparent native-to-ascii conversion属性主要用于转换ascii，不然Properties文件的中文会被转码，IntelliJ IDEA除了支持对整个Project设置编码之外
备份，不然可能出现转换过程变成乱码，无法还原。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，则会弹出演示

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。个人编码配置如下图所示：



类和方法注释模板

1.修改类注释模板

在File->Settings->Editor->File and Code Templates下分别修改Class，Interface，Enum等注释模板，Class模板部分修改如下，其余的举一反三进行修

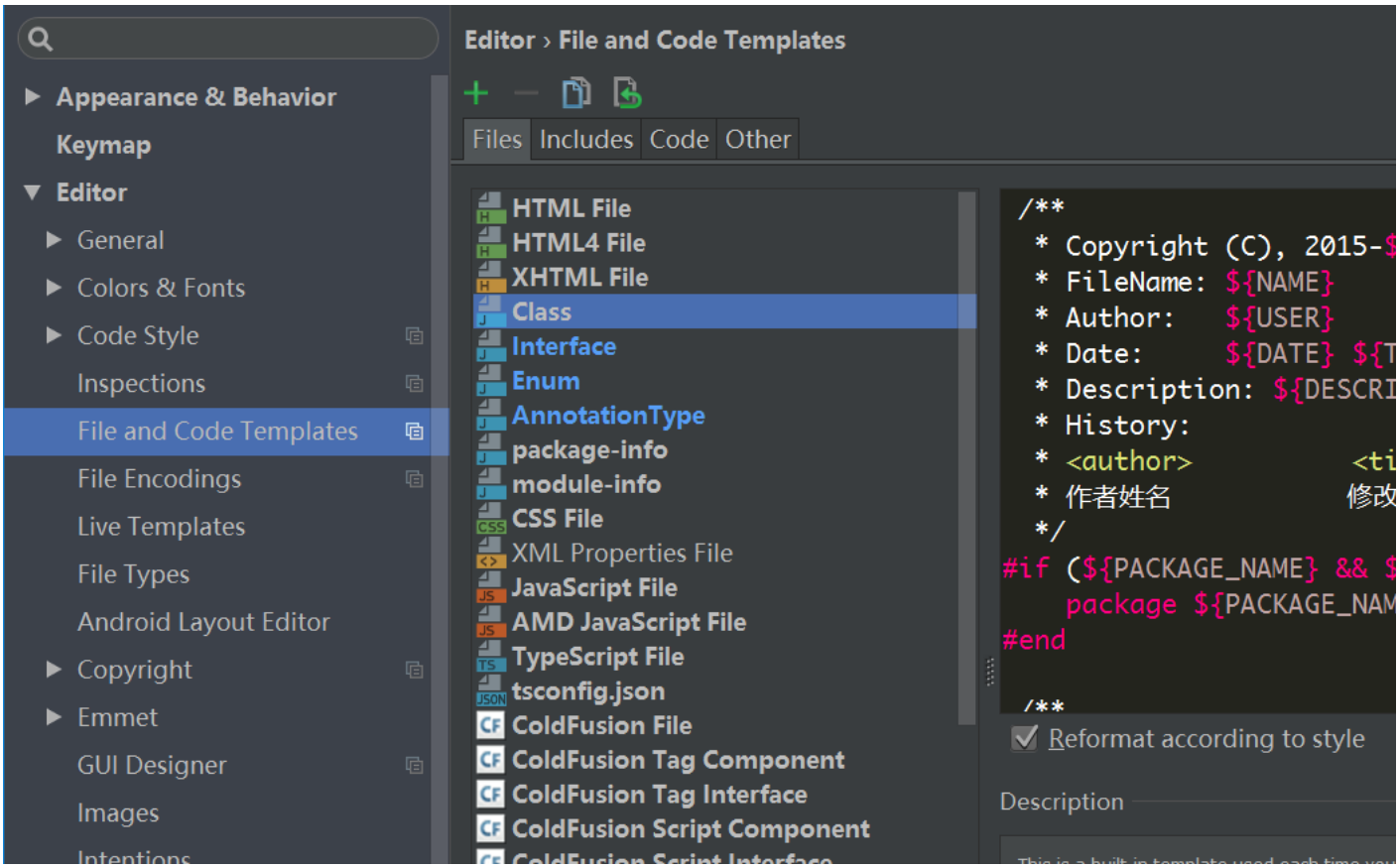
```
/**
 * Copyright (C), 2015-${YEAR}, XXX有限公司
 * FileName: ${NAME}
 * Author:   ${USER}
 * Date:     ${DATE} ${TIME}
 * Description: ${DESCRIPTION}
 * History:
 * <author>          <time>          <version>          <desc>
 * 作者姓名           修改时间           版本号              描述
 */
#if (${PACKAGE_NAME} && ${PACKAGE_NAME} != "")
    package ${PACKAGE_NAME};
#end

/**
 * 〈一句话功能简述〉<br>
 * 〈${DESCRIPTION}〉
 *
 * @author ${USER}
 * @create ${DATE}
 * @since 1.0.0
```

```
 */
public class ${NAME} {

}
```

类注释模板修改配置图



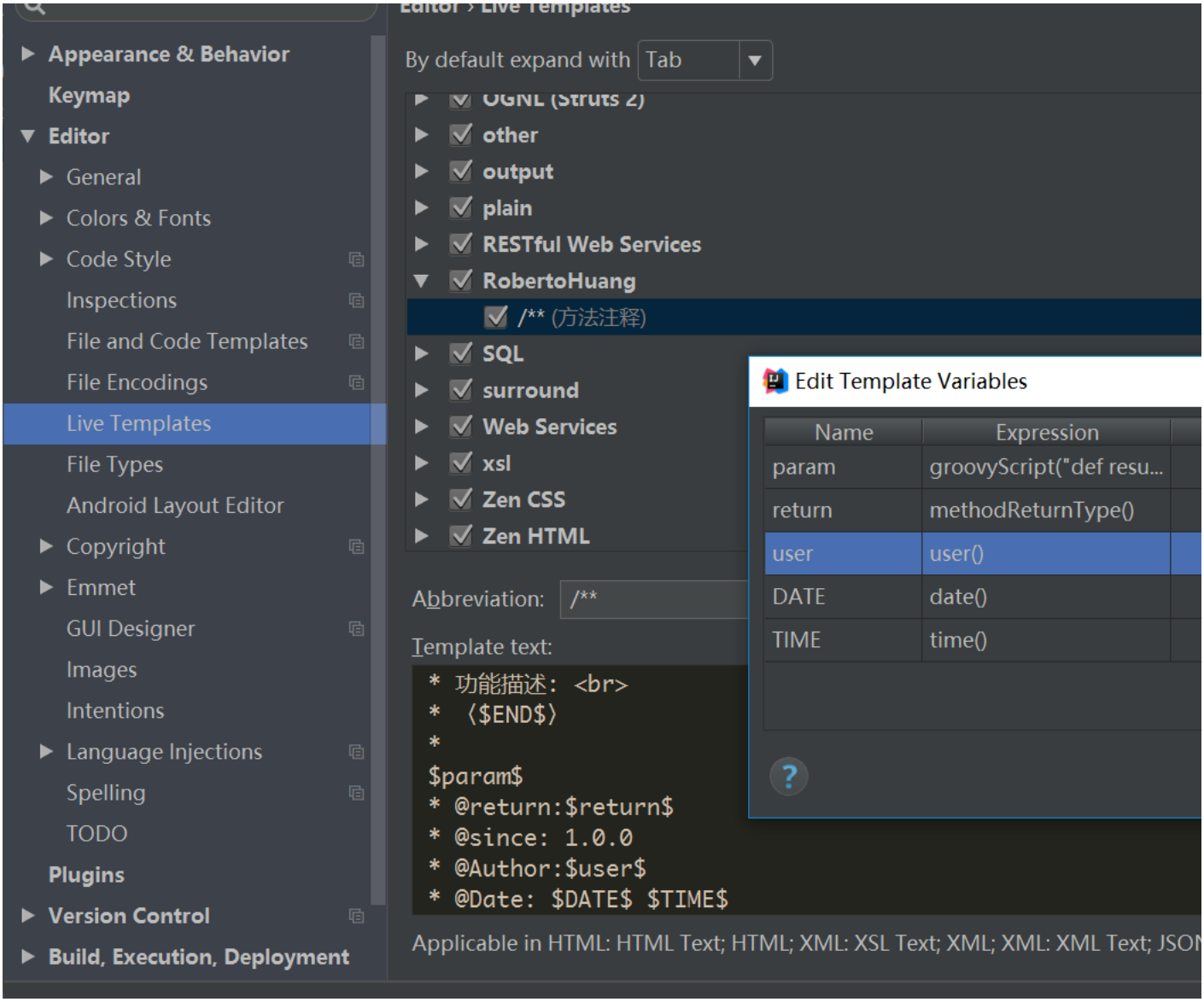
类注释模板修改效果图

```
/**
 * Copyright (C), 2015-2017, XXX有限公司
 * FileName: StudentService
 * Author:   RobertoHuang
 * Date:     2017/7/13 22:00
 * Description: 与学生有关Service
 * History:
 * <author>          <time>          <version>          <desc>
 * 作者姓名           修改时间          版本号              描述
 */
package com.roberto;

/**
 * 〈一句话功能简述〉<br>
 * 〈与学生有关Service〉
 *
 * @author RobertoHuang
 * @create 2017/7/13
 * @since 1.0.0
 */
public interface StudentService {
}
```


2.方法注释模板修改

在File->Settings->Editor->Live Templates下添加自定义Template Group，并在自定义Template Group下添加自定义Template，具体配置如下图所示



详细配置参数

```
Template text内容如下:
/**
 * 功能描述: <br>
 * (&#36;END&#36;)
 *
 &#36;param&#36;
 * @return:&#36;return&#36;
 * @since: 1.0.0
 * @Author:&#36;user&#36;
 * @Date: &#36;DATE&#36; &#36;TIME&#36;
 */

Edit Template Variables请求参数部分内容如下:
groovyScript("def result=''; def params=\"${_1}\".replaceAll('[\\\\[|\\\\]|\\\\s]', '').split(',').toList(); for(i = 0; i <
```

```
13
14
```

在完成如上配置后，只需在方法内执行/**+Enter键即可生成注释，切记这里说的是方法内部，因为methodParameters()的作用域只在方法内部，这也是

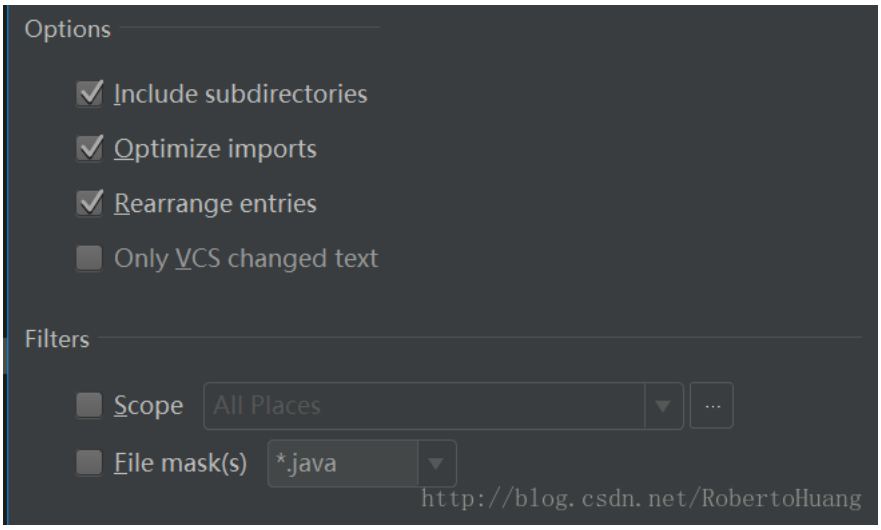
下

方法注释模板修改效果图

```
public static void main(String args[]) {
    /**
     * 功能描述: <br>
     * 〈〉
     *
     * @param args
     * @return:void
     * @since: 1.0.0
     * @Author:RobertoHuang
     * @Date: 2017/7/13 22:06
     */
}
```


代码格式化

代码格式化的快捷键为Ctrl+Alt+L，如果在类中执行代码格式化则会对代码进行排版，若焦点在类或者文件夹上，则会弹出格式化选项提示框，弹出框如

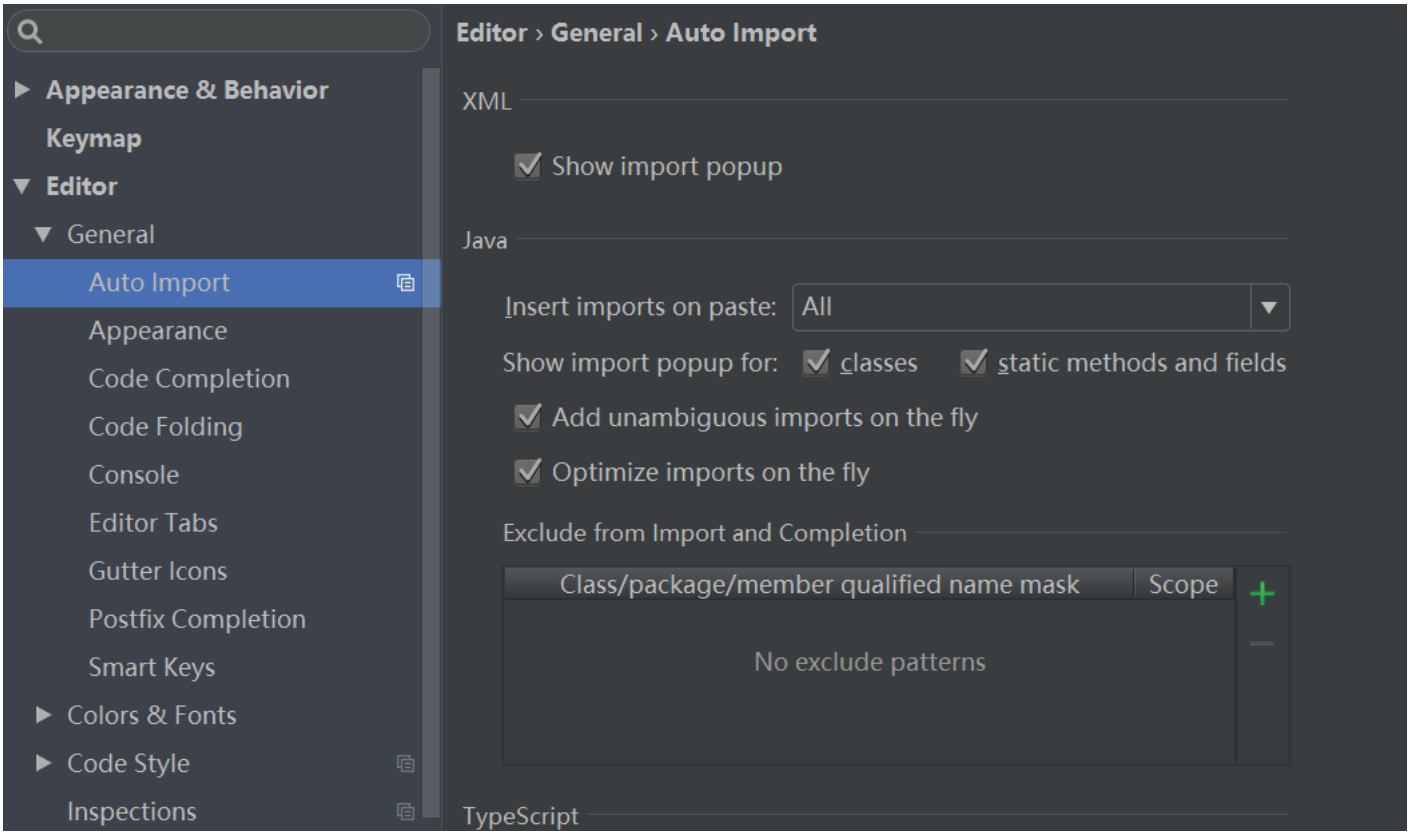



```
1 Include subdirectories:是否对子目录也进行格式化
2 Optimize imports:优化导入的类和包
3 Rearrange enries:对代码顺序进行调整(将Filed放在Method前边)
4
5 Filters即配置过滤条件，表示对哪些文件进行格式化
```

```
1
2
3
4
5
```

自动导入所有包

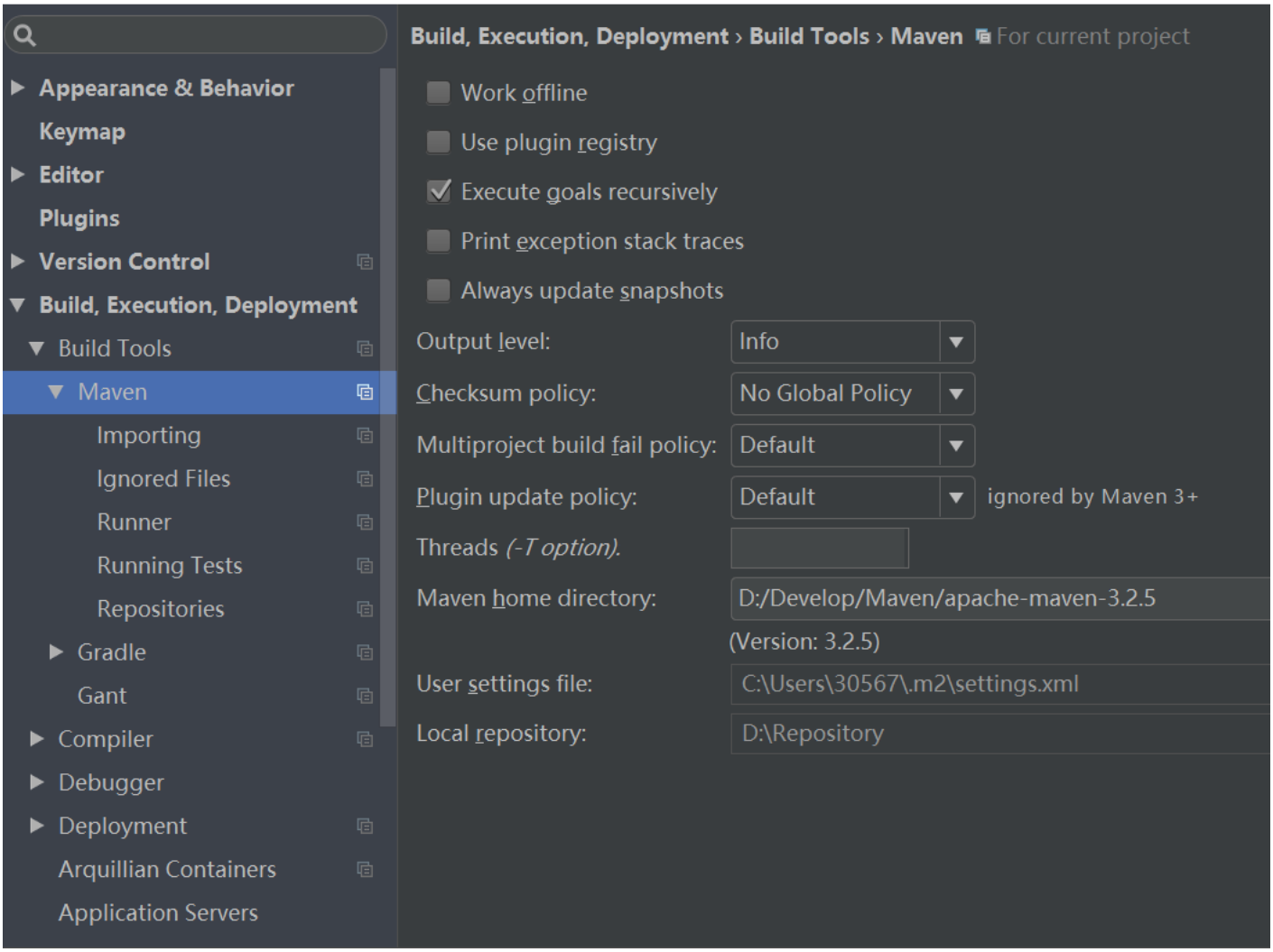
在Intellij IDEA一次只能导入单个包，没有像Eclipse快速导入包的快捷键Ctrl+Shift+O，但是Intellij IDEA下有个自动导入包的功能。在File->Settings->E



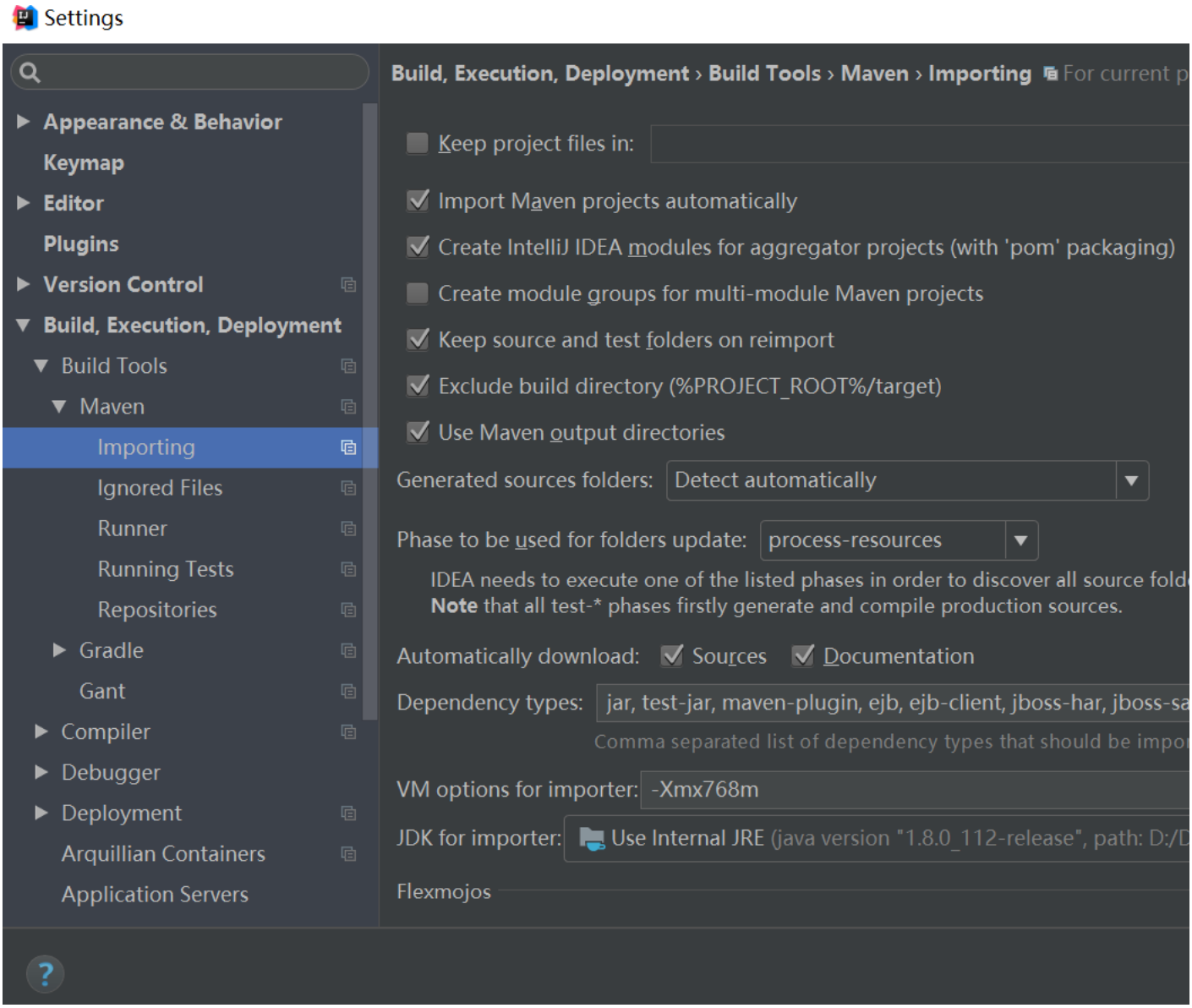

```
Insert imports on paste:复制代码的时候，对于导入的包是否需要进行询问的一个选项。
    ASK(有需要导入的包名时会弹提示框，问你要不要导入)
    NONE(有需要导入的包名时不会弹提示框，也不会自动导入)
    ALL(有需要导入的包名时会自动导入，不会弹提示框)
Show import popup:当输入的类的声明没被导入时，会弹出一个选择的对话框
Optimize imports on fly:自动优化包导入，移除不需要的包
Add unambiguous imports on the fly:这个就是自动导入功能了，当你输入类名后声明就被自动导入了
Exclude from Import and Completion:这个其实就是你自定义import,可以不用关注，一般来说你是用不上的
```

Maven配置

在File->Settings->Build,Execution,Deployment->Build Tools->Maven下对Maven进行配置，个人配置如下图所示




```
user settings file:指定Maven的settings.xml位置
local repository: 指定Maven的本地仓库位置，是读取settings.xml自动配置的
maven home directory:指定本地Maven的安装目录所在，因为我已经配置了MAVEN_HOME系统参数，所以直接这样配置IntelliJ IDEA 是可以找到的，但
```

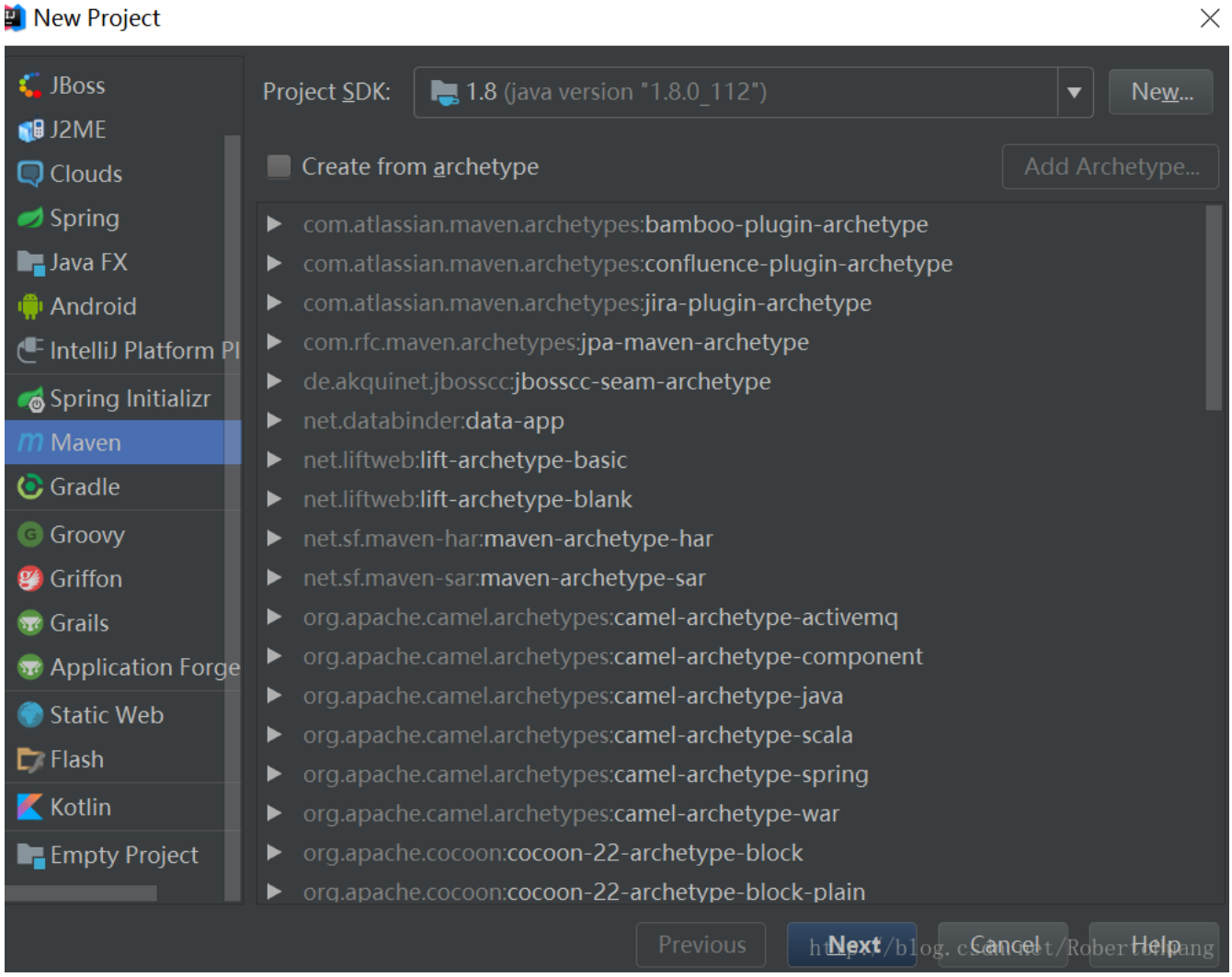


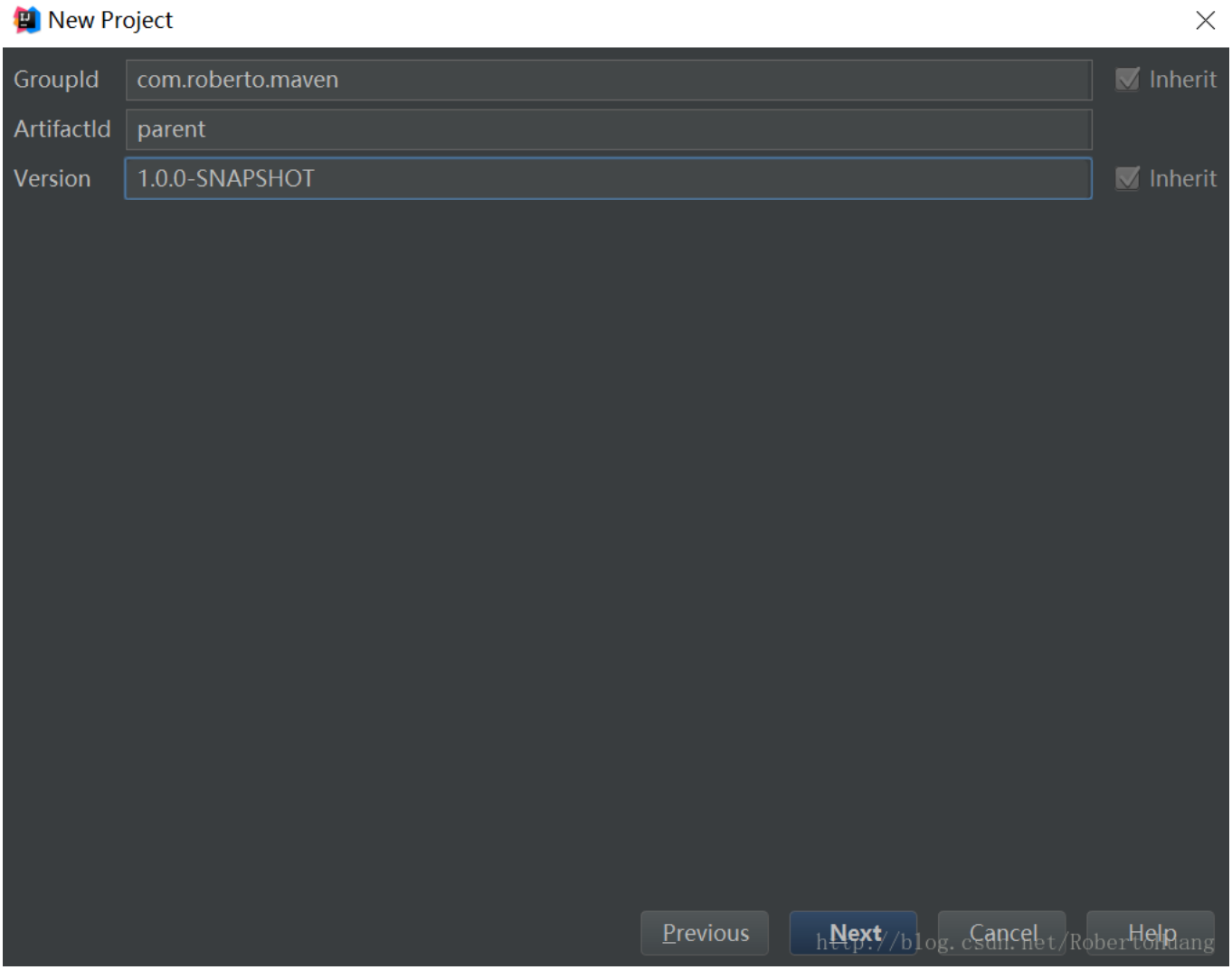

```
1 VM options for importer:可以设置导入的VM参数，一般这个都不需要主动改，除非项目真的导入太慢了我们再增大此参数
2 Import Maven projects automatically:表示IntelliJ IDEA会实时监控项目的pom.xml文件进行项目变动设置，建议进行勾选
3 Sources和Documentation:表示在Maven导入依赖包的时候是否自动下载源码和文档，默认是没有勾选的也不建议勾选，原因是这样可以加快项目从外网导
```

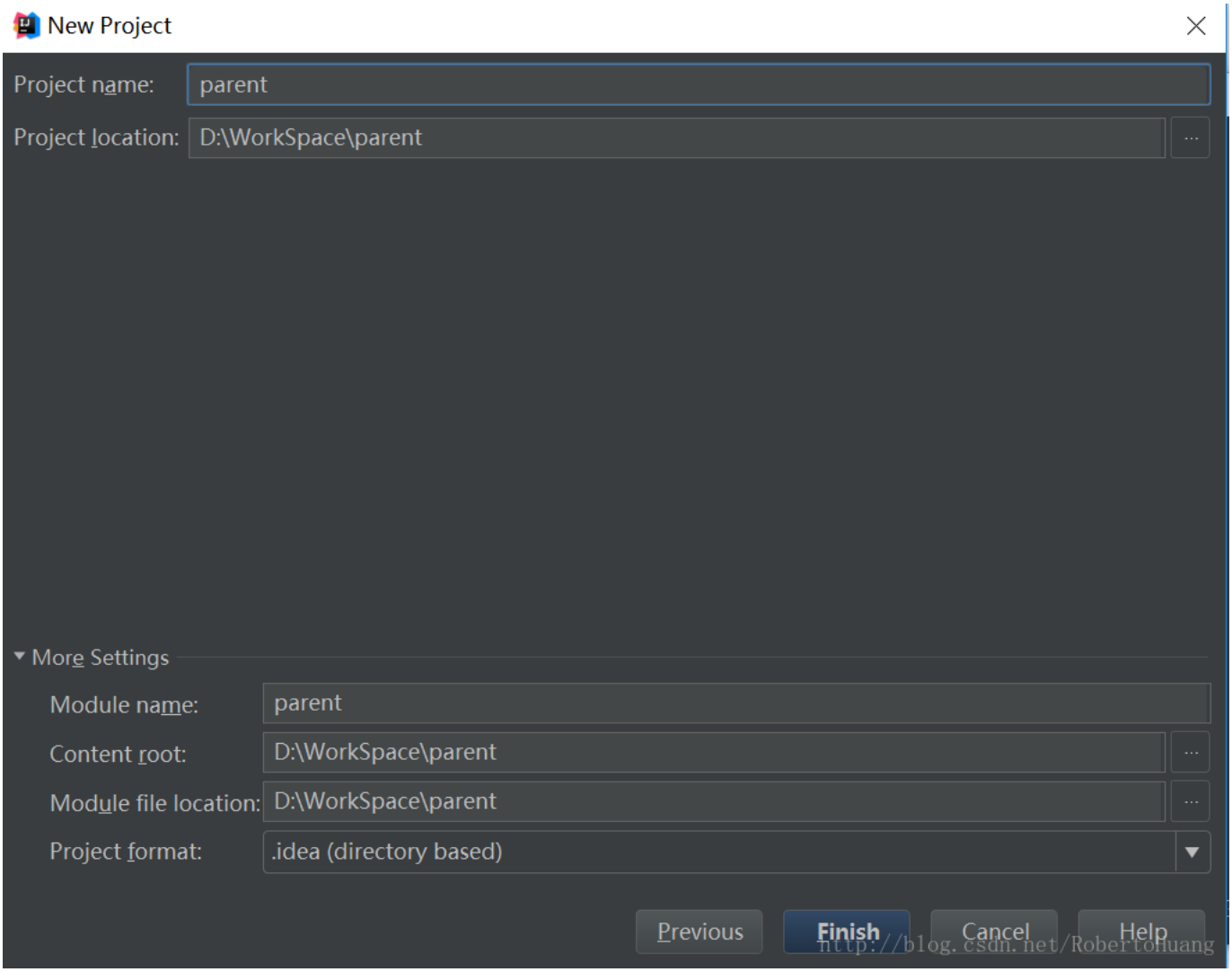
```
1
2
3
```

Maven聚合工程搭建

1.创建父工程(不需要使用模板)

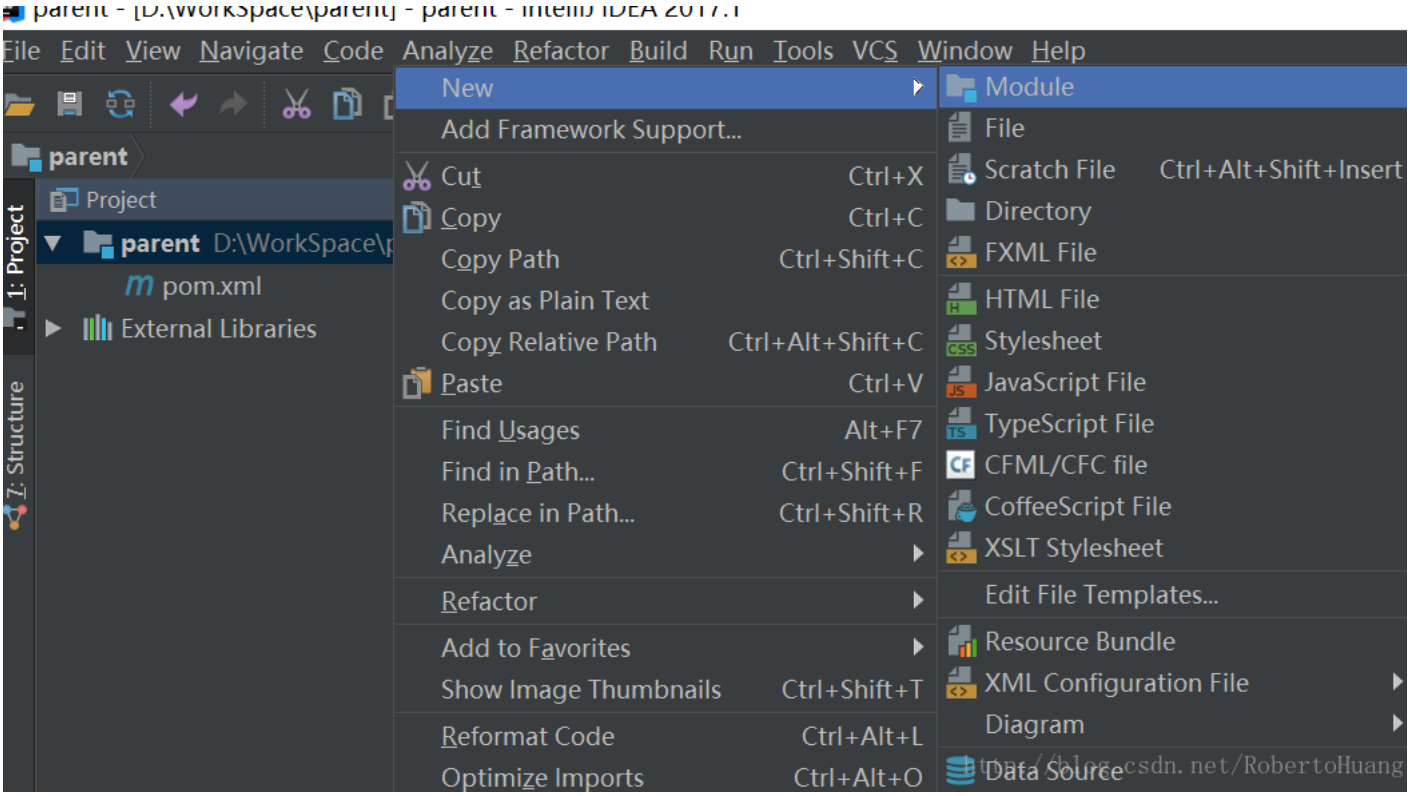


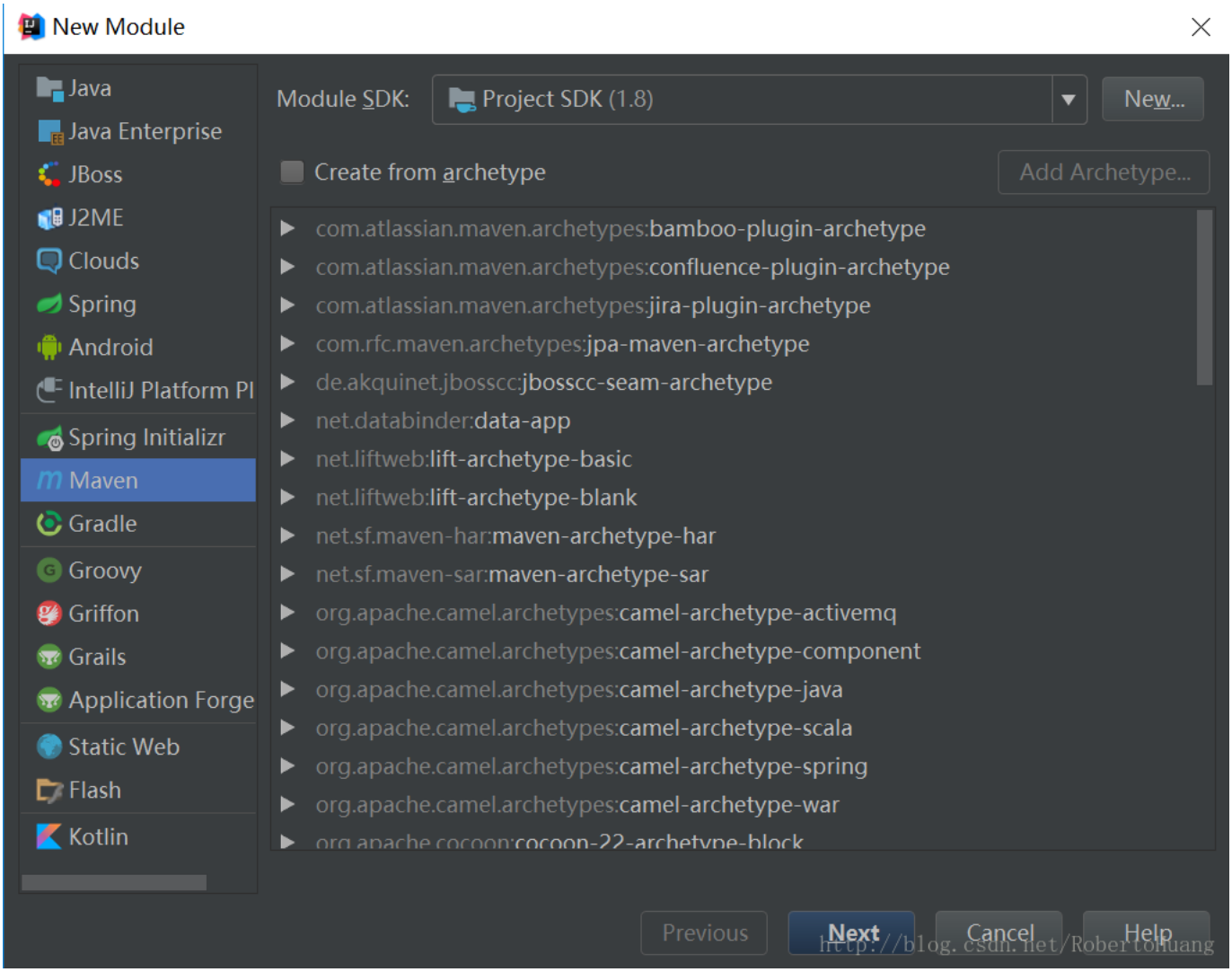

New Project
GroupId
com.roberto.maven
Inherit
ArtifactId
parent
Version
1.0.0-SNAPSHOT
Inherit
Previous
Next
Cancel
Help
http://blog.csdn.net/Robertohuang

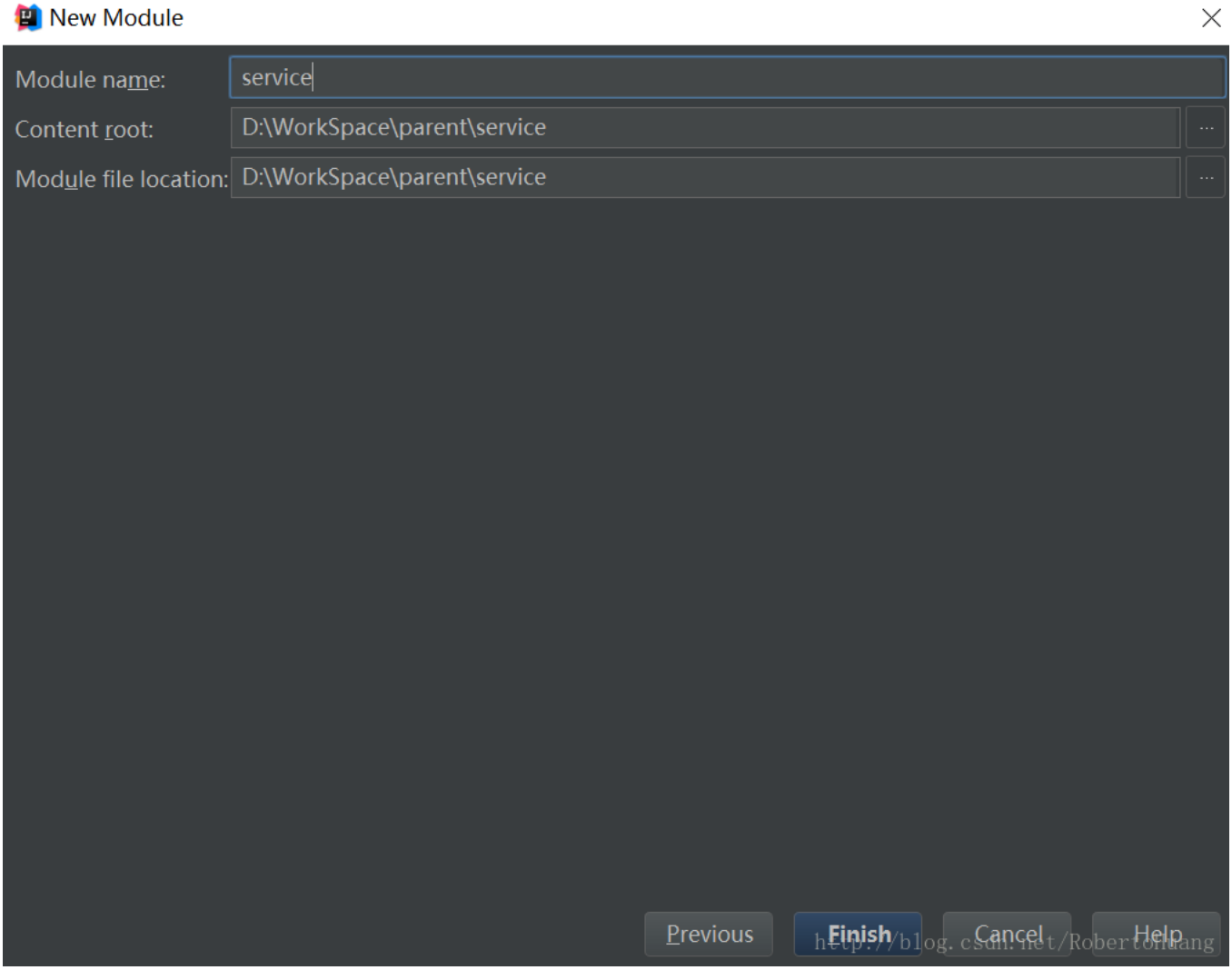


创建完父工程后删除父工程的src目录，该目录在聚合项目中无用

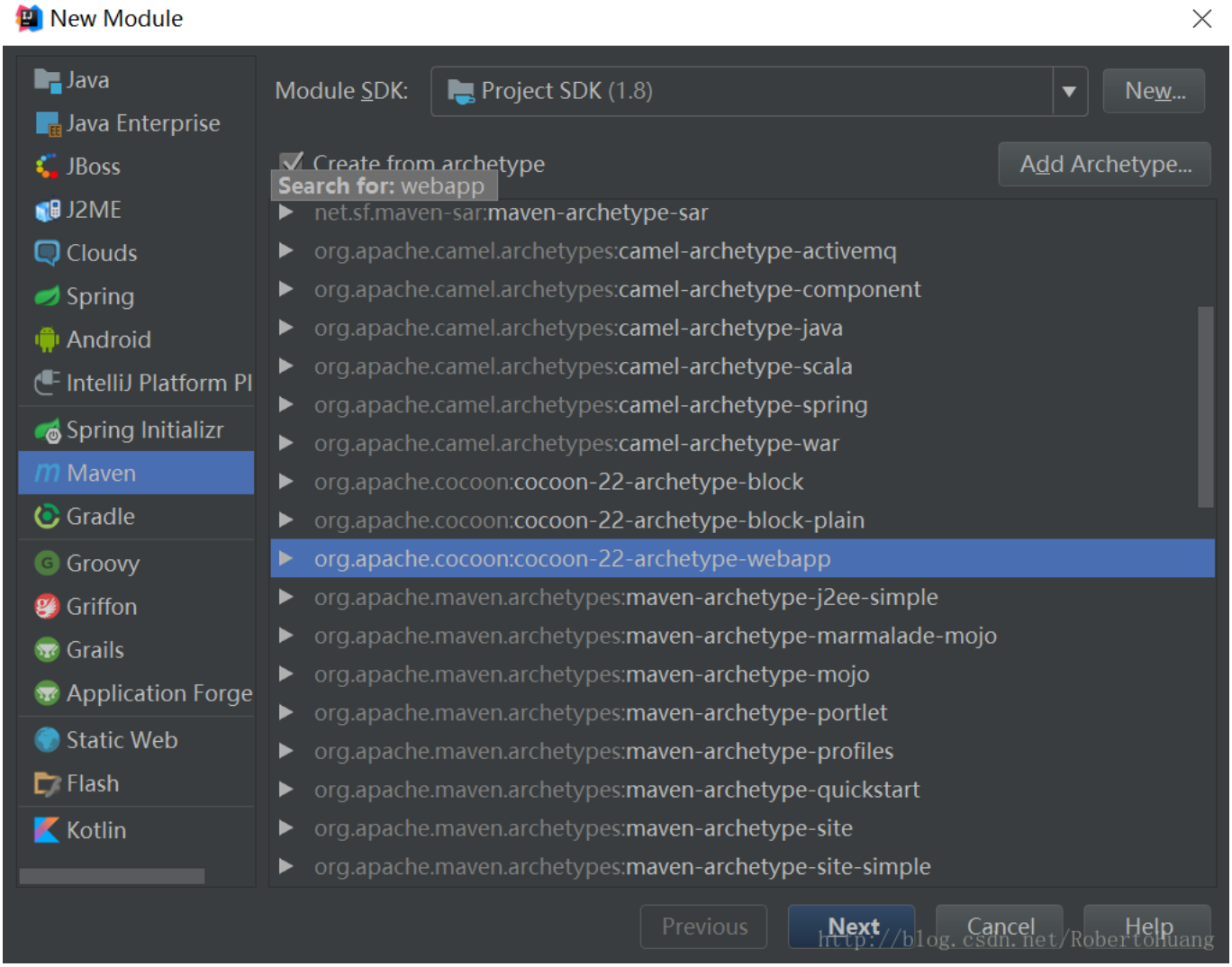
2.添加非web子模块

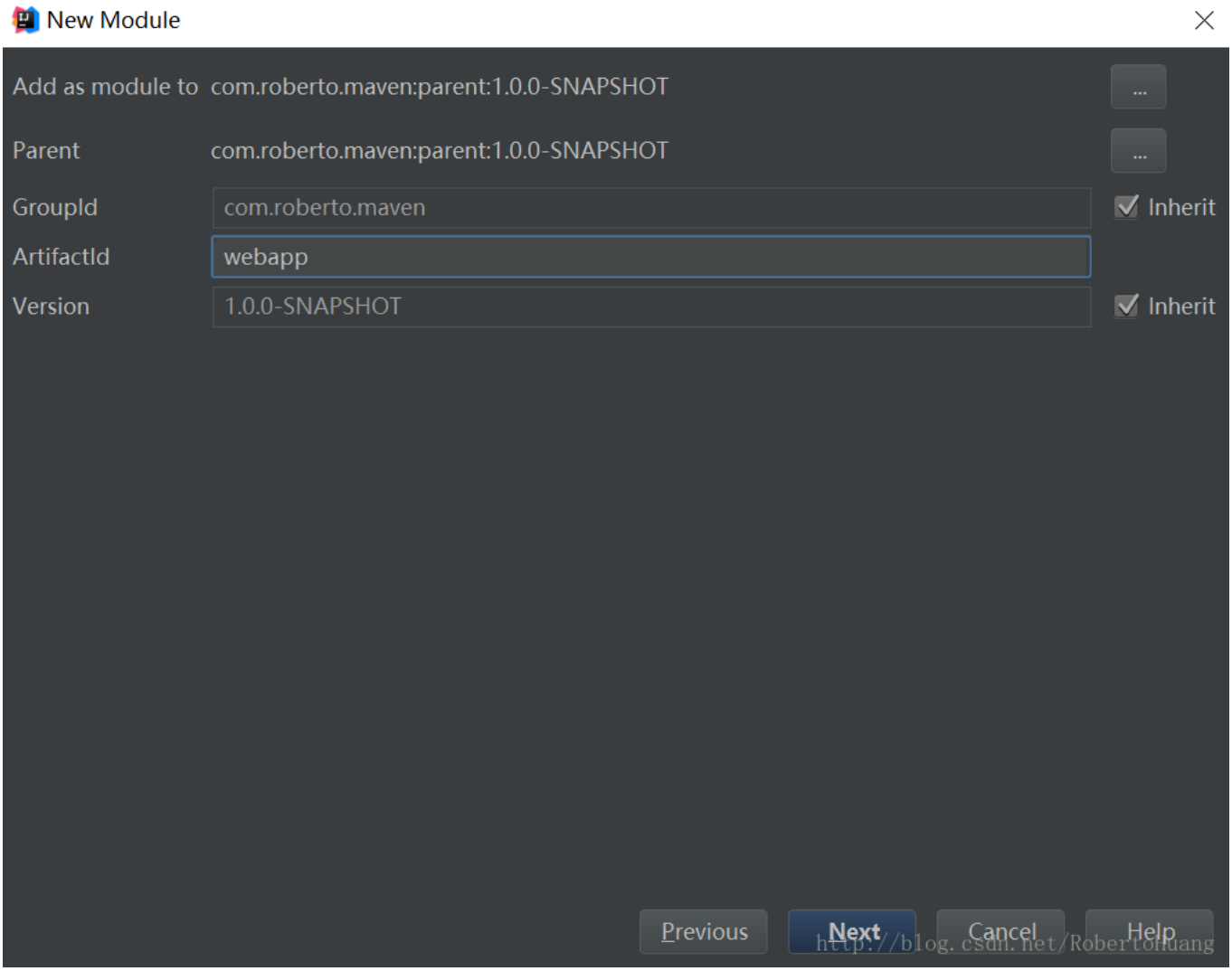


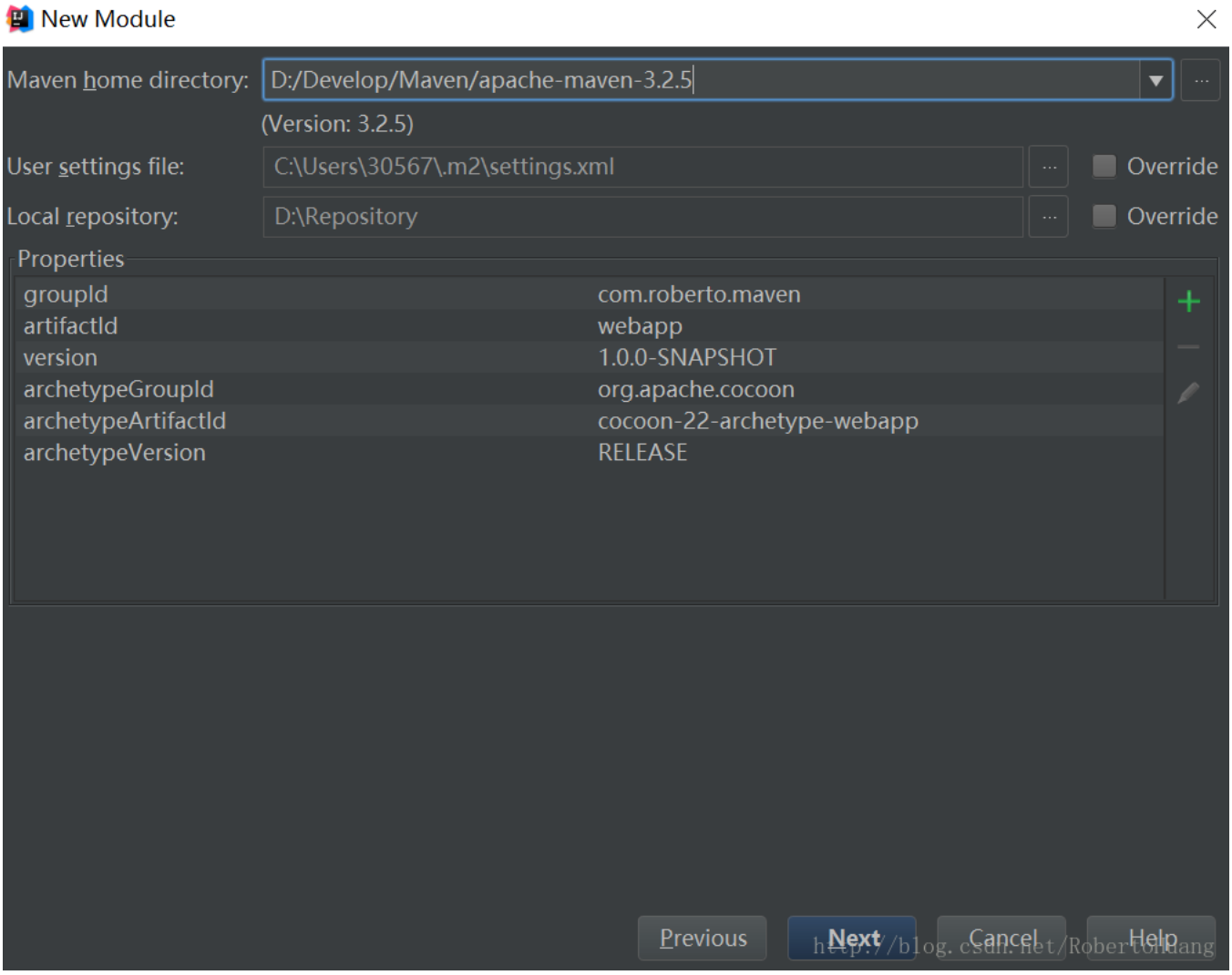

New Module
Java
Java Enterprise
JBoss
J2ME
Clouds
Spring
Android
IntelliJ Platform Pl
Spring Initializr
Maven
Gradle
Groovy
Griffon
Grails
Application Forge
Static Web
Flash
Kotlin
Module SDK:
Project SDK (1.8)
New...
Create from archetype
Add Archetype...
com.atlassian.maven.archetypes:bamboo-plugin-archetype
com.atlassian.maven.archetypes:confluence-plugin-archetype
com.atlassian.maven.archetypes:jira-plugin-archetype
com.rfc.maven.archetypes:jpa-maven-archetype
de.akquinet.jbosscc:jbosscc-seam-archetype
net.databinder:data-app
net.liftweb:lift-archetype-basic
net.liftweb:lift-archetype-blank
net.sf.maven-har:maven-archetype-har
net.sf.maven-sar:maven-archetype-sar
org.apache.camel.archetypes:camel-archetype-activemq
org.apache.camel.archetypes:camel-archetype-component
org.apache.camel.archetypes:camel-archetype-java
org.apache.camel.archetypes:camel-archetype-scala
org.apache.camel.archetypes:camel-archetype-spring
org.apache.camel.archetypes:camel-archetype-war
Previous
Next
Cancel
Help
http://blog.csdn.net/Robertohuang

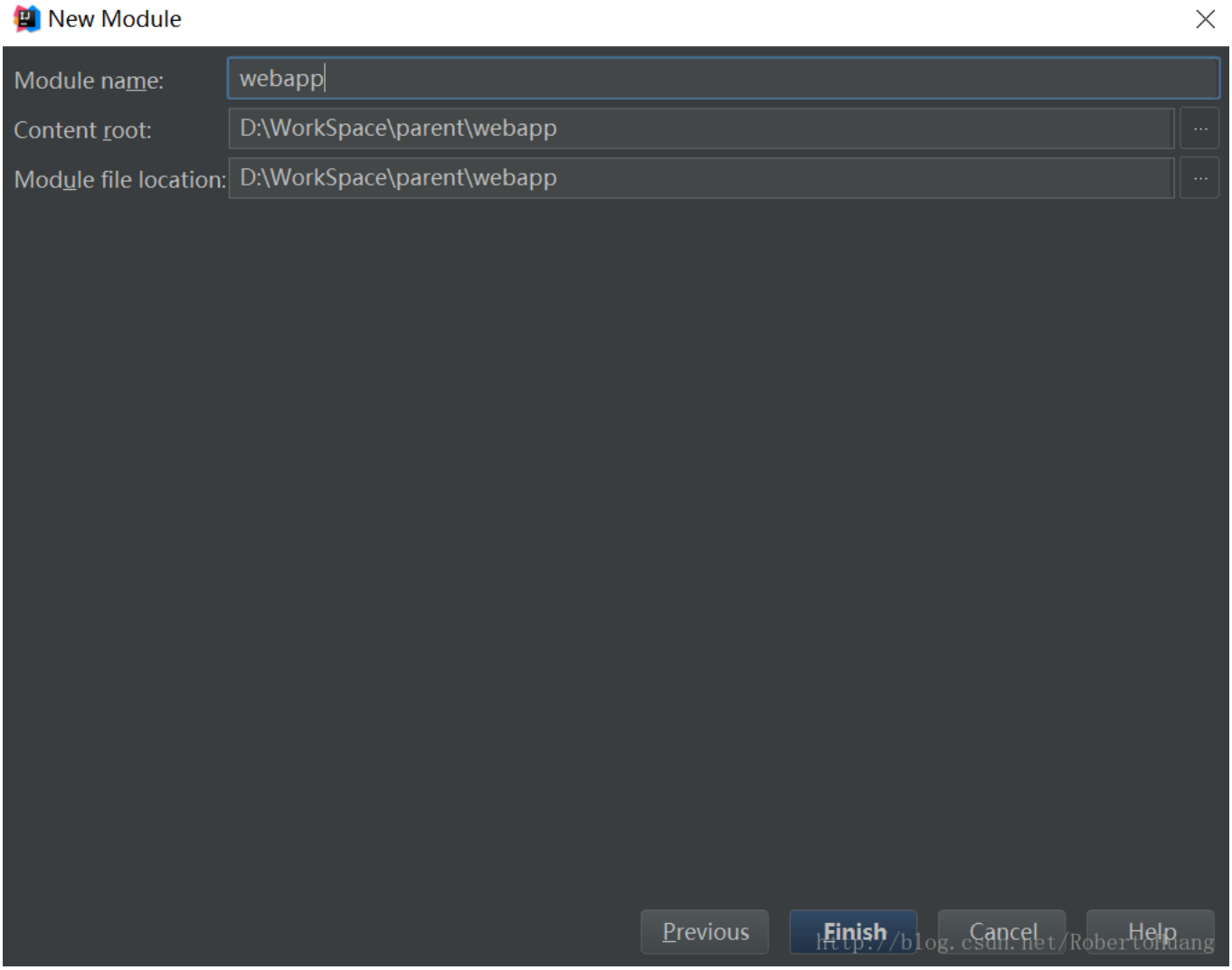

New Module
Module name:
service
Content root:
D:\WorkSpace\parent\service
Module file location:
D:\WorkSpace\parent\service
Previous
Finish
Cancel
Help

3.添加web子模块(使用maven web项目模板)

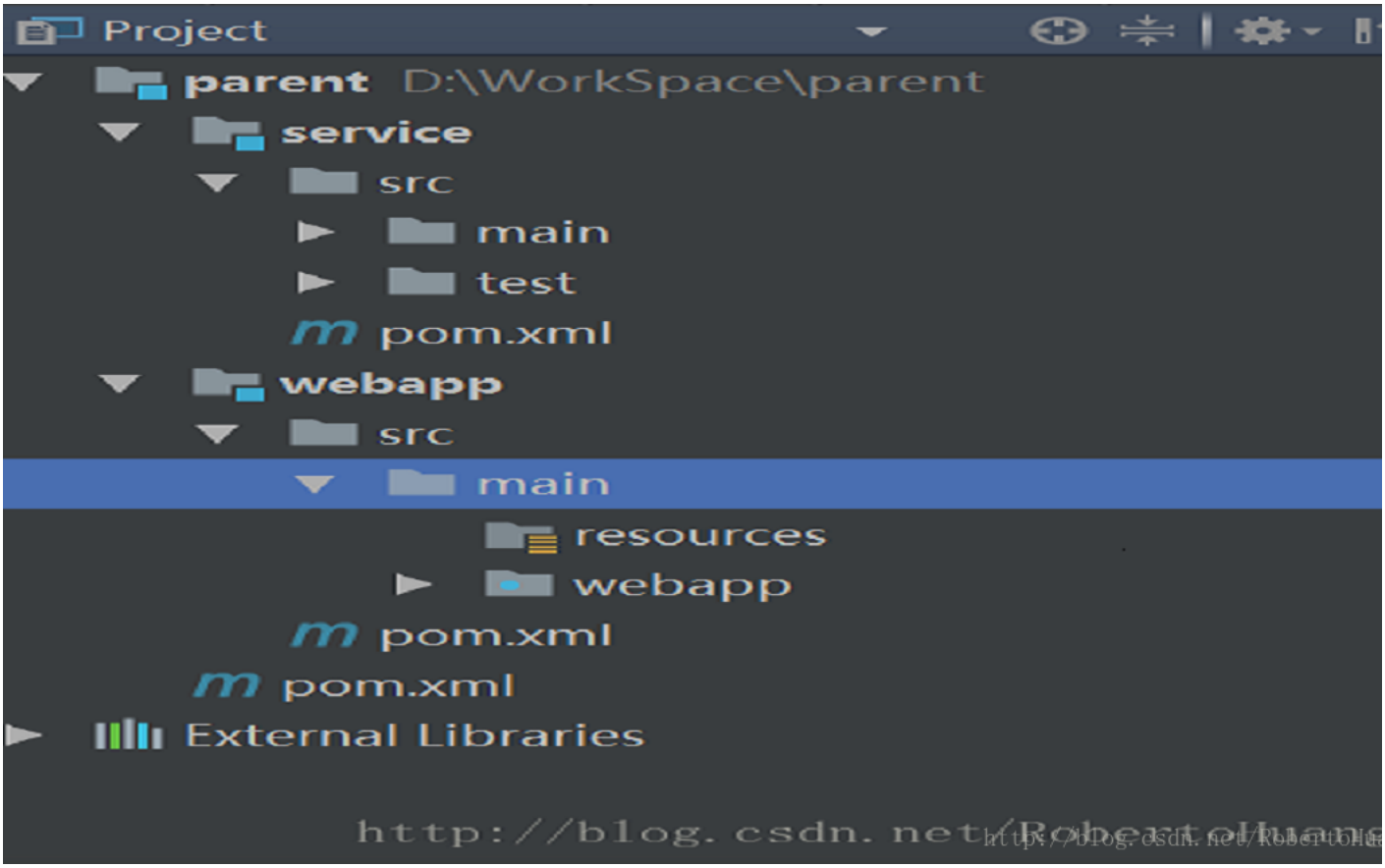



New Module
Add as module to com.roberto.maven:parent:1.0.0-SNAPSHOT
...
Parent com.roberto.maven:parent:1.0.0-SNAPSHOT
...
GroupId com.roberto.maven
Inherit
ArtifactId webapp
Version 1.0.0-SNAPSHOT
Inherit
Previous
Next
Cancel
Help
http://blog.csdn.net/Robertohuang


New Module
Maven home directory:
D:/Develop/Maven/apache-maven-3.2.5
(Version: 3.2.5)
User settings file:
C:\Users\30567\.m2\settings.xml
Override
Local repository:
D:\Repository
Override
Properties
groupId
com.roberto.maven
artifactId
webapp
version
1.0.0-SNAPSHOT
archetypeGroupId
org.apache.cocoon
archetypeArtifactId
cocoon-22-archetype-webapp
archetypeVersion
RELEASE
Previous
Next
Cancel
Help



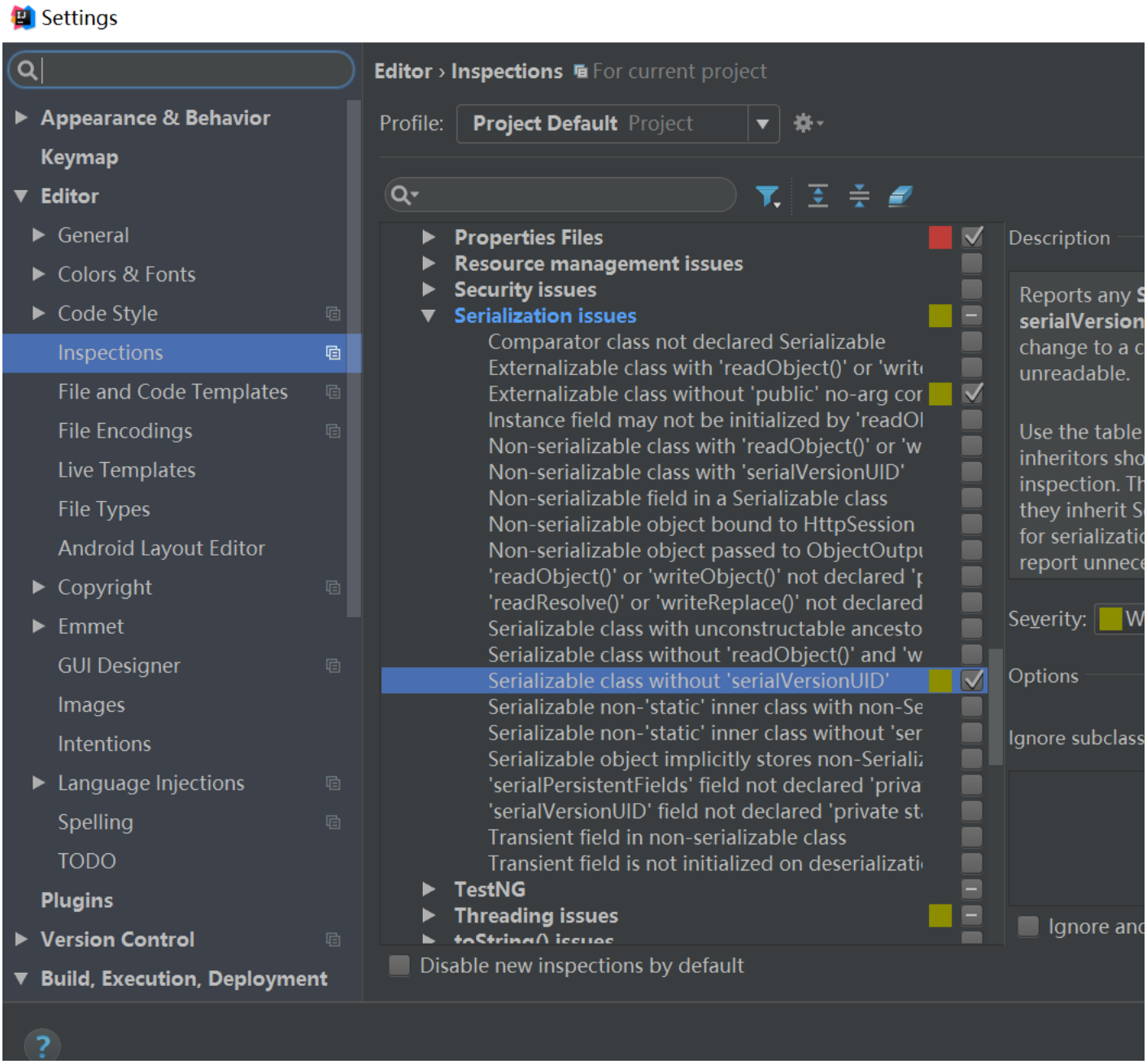
4.创建完查看工程目录结构如下



5.打开pom.xml文件 观察pom.xml文件的变化

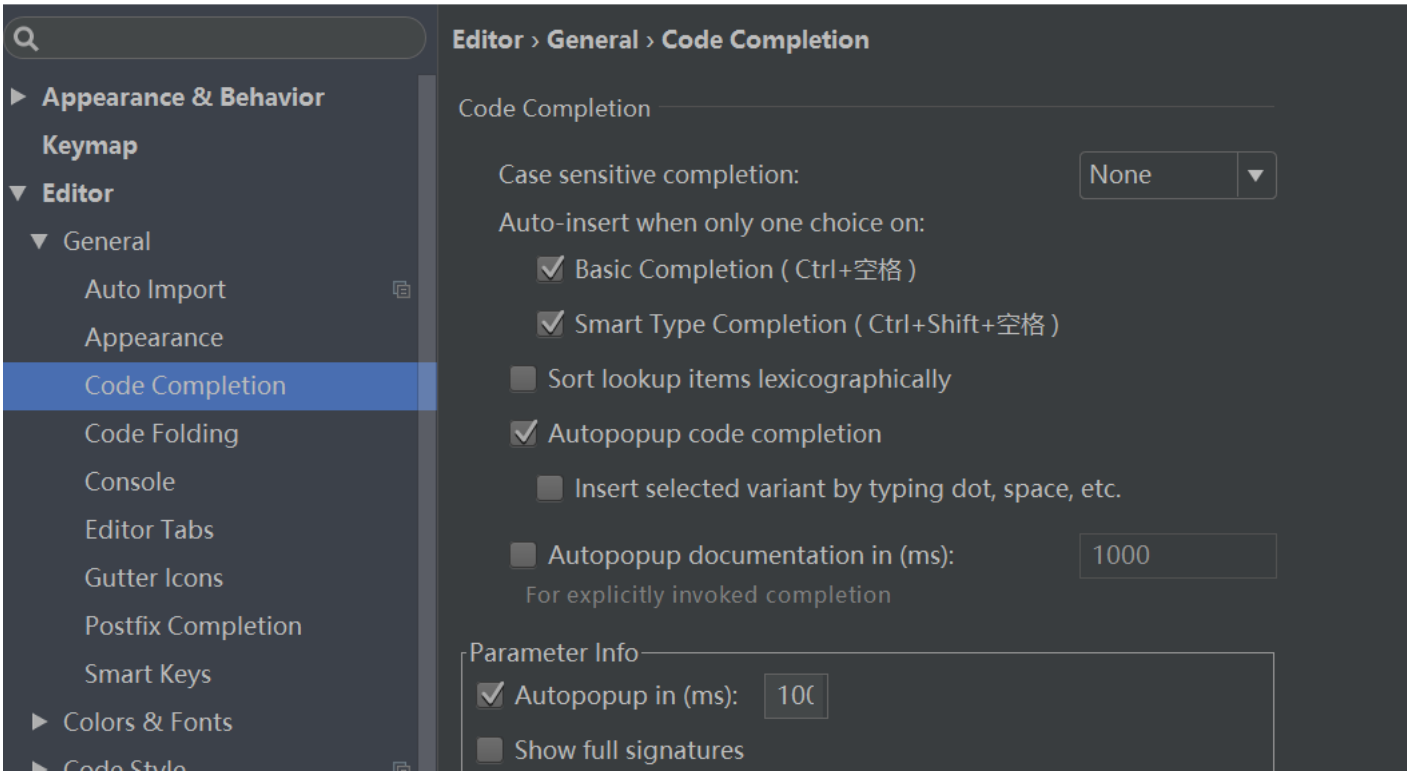
生成serialVersionUID

默认情况下Intellij IDEA关闭了继承了Java.io.Serializable的类生成serialVersionUID的警告，如果需要提示生成serialVersionUID，那么需要做以下设置键就会提示生成serialVersionUID了


Settings
Appearance & Behavior
Keymap
Editor
General
Colors & Fonts
Code Style
Inspections
File and Code Templates
File Encodings
Live Templates
File Types
Android Layout Editor
Copyright
Emmet
GUI Designer
Images
Intentions
Language Injections
Spelling
TODO
Plugins
Version Control
Build, Execution, Deployment
Editor › Inspections For current project
Profile: Project Default Project
Properties Files
Resource management issues
Security issues
Serialization issues
Comparator class not declared Serializable
Externalizable class with 'readObject()' or 'writ
Externalizable class without 'public' no-arg cor
Instance field may not be initialized by 'readOl
Non-serializable class with 'readObject()' or 'w
Non-serializable class with 'serialVersionUID'
Non-serializable field in a Serializable class
Non-serializable object bound to HttpSession
Non-serializable object passed to ObjectOutpu
'readObject()' or 'writeObject()' not declared 'p
'readResolve()' or 'writeReplace()' not declared
Serializable class with unconstructable ancesto
Serializable class without 'readObject()' and 'w
Serializable class without 'serialVersionUID'
Serializable non-'static' inner class with non-Se
Serializable non-'static' inner class without 'ser
Serializable object implicitly stores non-Seriali
'serialPersistentFields' field not declared 'priva
'serialVersionUID' field not declared 'private st
Transient field in non-serializable class
Transient field is not initialized on deserializati
TestNG
Threading issues
Description
Reports any S
serialVersion
change to a c
unreadable.
Use the table
inheritors sho
inspection. Th
they inherit S
for serializatio
report unnece
Severity:
Options
Ignore subclass
Ignore and
Disable new inspections by default

代码提示忽略大小写

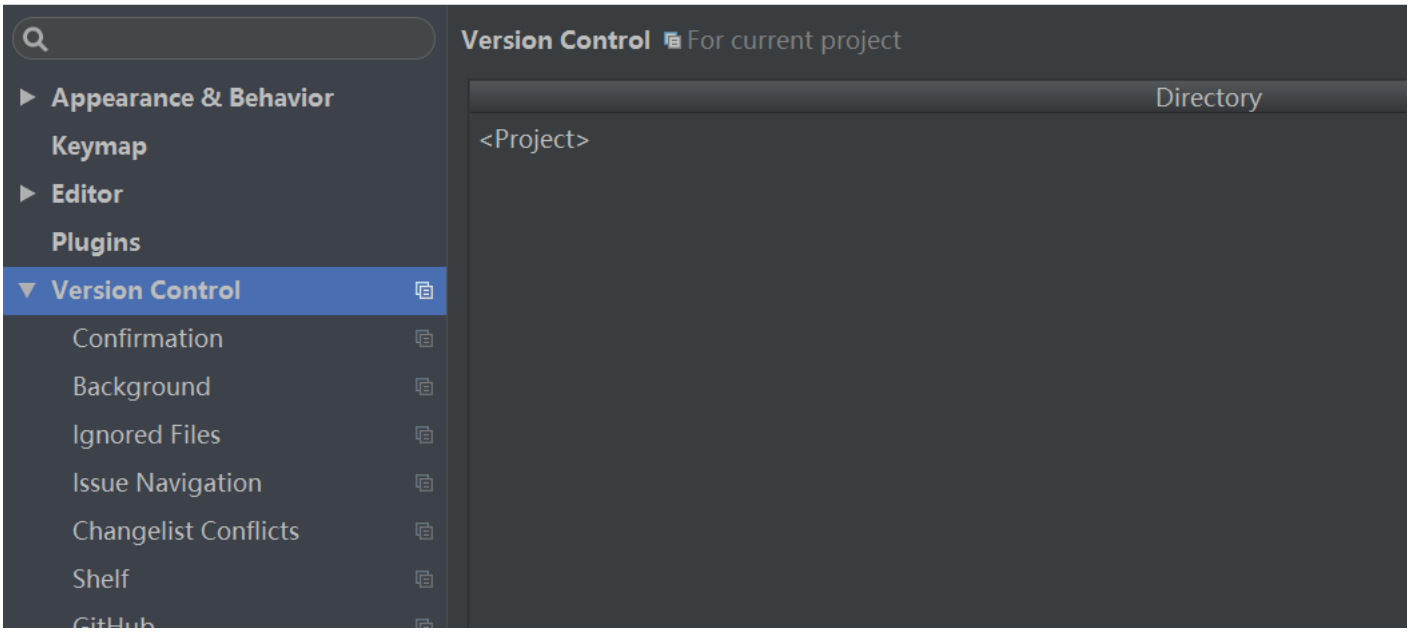
在File->Settings->Editor->General->Code Completion下设置Case sensitive completion为none



IDEA脱离版本控制

现在版本控制都有对应的优秀免费的开源客户端，而且稳定性更好。更多时候我们希望IDEA只是作为开发工具使用，而不参与版本控制，在File->Sett

Settings



实用插件推荐

快捷键提示插件

Key promoter是在你通过非快捷键方式使用某功能时 为你提供快捷键建议 在开始记不住快捷键的情况下 强烈推荐安装

翻译插件

翻译插件 TranslationPlugin,支持支持中英互译、单词朗读,详细安装文档请参考:TranslationPlugin介绍与安装手册

热部署插件JRebel

JRebel热部署插件安装和使用请参考:JRebel热部署插件安装和使用

Maven Helper

Maven 辅助插件 用于查找Maven依赖冲突非常好用的一款插件 安装步骤请参考:Maven Helper安装使用

Properties to YAML Converter

在开发SpringBoot项目时，会需要把Properties的配置格式改为 YAML格式，Properties to YAML Converter提供了很好的支持

阿里巴巴代码规范插件p3c-pmd

详细安装和使用请参考:阿里巴巴代码规范插件p3c-pmd

开发必备快捷键

IntelliJ IDEA提供了丰富的快捷键组合来加快开发效率，但是快捷键太多琳琅满目也会给人无从下手的感觉。下面是我个人整理的在开发过程中必备的Features Trainer:https://plugins.jetbrains.com/plugin/8554?pr=idea，大家可以自行去插件库下载学习

Ctrl相关

快捷键	介绍
Ctrl + B	进入光标所在的方法/变量的接口或是定义处，等效于Ctrl + 左键单击
Ctrl + D	复制光标所在行或复制选择内容，并把复制内容插入光标位置下面
Ctrl + F	在当前文件进行文本查找
Ctrl + H	查看类的继承结构
Ctrl + N	通过类名定位文件
Ctrl + O	快速重写父类方法
Ctrl + P	方法参数提示
Ctrl + Y	删除光标所在行或删除选中的行
Ctrl + W	递进式选择代码块
Ctrl + Z	撤销
Ctrl + 1,2,3…9	定位到对应数值的书签位置 结合Ctrl + Shift + 1,2,3…9使用
Ctrl + F1	在光标所在的错误代码出显示错误信息
Ctrl + F12	弹出当前文件结构层，可以在弹出的层上直接输入进行筛选
Ctrl + Space	基础代码补全默认在Windows系统上被输入法占用，需要进行修改，建议修改为Ctrl + 逗号
Ctrl + /	注释光标所在行代码，会根据当前不同文件类型使用不同的注释符号

Alt相关

快捷键	介绍
Alt + Q	弹出一个提示，显示当前类的声明/上下文信息
Alt + Enter	根据光标所在问题，提供快速修复选择

Shift相关

快捷键	介绍
Shift + F3	在查找模式下，定位到上一个匹配处

Ctrl+Alt相关

快捷键	介绍
Ctrl + Alt + B	在某个调用的方法名上使用会跳到具体的实现处

快捷键	介绍
Ctrl + Alt + L	格式化代码 可以对当前文件和整个包目录使用
Ctrl + Alt + M	快速抽取方法
Ctrl + Alt + O	优化导入的类和包 可以对当前文件和整个包目录使用
Ctrl + Alt + T	对选中的代码弹出环绕选项弹出层
Ctrl + Alt + V	快速引进变量
Ctrl + Alt + F7	寻找类或是变量被调用的地方，以弹出框的方式显示
Ctrl + Alt + 左方向键	退回到上一个操作的地方
Ctrl + Alt + 右方向键	前进到上一个操作的地方

Ctrl+Shift相关

快捷键	介绍
Ctrl + Shift + F	根据输入内容查找整个项目或指定目录内文件
Ctrl + Shift + H	查看方法的继承结构
Ctrl + Shift + J	自动将下一行合并到当前行末尾
Ctrl + Shift + N	通过文件名定位打开文件/目录，打开目录需要在输入的内容后面多加一个正斜杠
Ctrl + Shift + R	根据输入内容替换对应内容，范围为整个项目或指定目录内文件
Ctrl + Shift + U	对选中的代码进行大/小写轮流转换
Ctrl + Shift + W	递进式取消选择代码块
Ctrl + Shift + Z	取消撤销
Ctrl + Shift + /	代码块注释
Ctrl + Shift + +	展开所有代码
Ctrl + Shift + -	折叠所有代码
Ctrl + Shift + 1,2,3…9	快速添加指定数值的书签
Ctrl + Shift + F7	高亮显示所有该选中文本，按Esc高亮消失
Ctrl + Shift + Space	智能代码提示
Ctrl + Shift + Enter	自动结束代码，行末自动添加分号

Alt+Shift相关

快捷键
快捷键

Ctrl+Alt+Shift相关

快捷键

其他

快捷键	介绍
F2	跳转到下一个高亮错误或警告位置
F3	在查找模式下，定位到下一个匹配处

快捷键	介绍
F4	编辑源