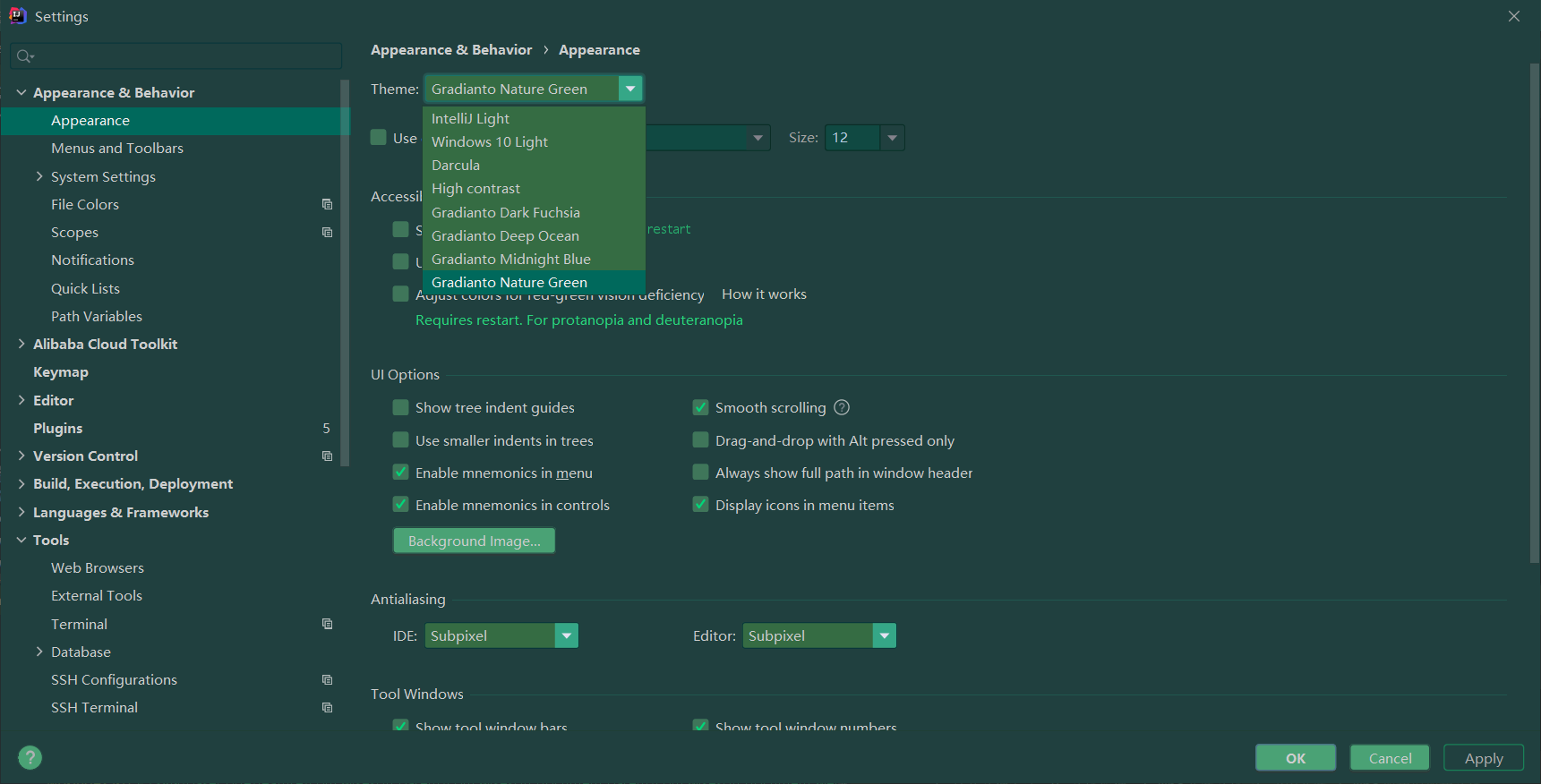
## 1.idea help官网

https://www.jetbrains.com/help/idea/

## 2.相关插件安装

### 2.1 Gradianto插件

支持idea的相关皮肤，是我用过比较好用，ui风格很好的一款插件



### 2.2 jclasslib Bytecode viewer插件

class文件反编译插件 ->jclasslib is a bytecode viewer for Java class files

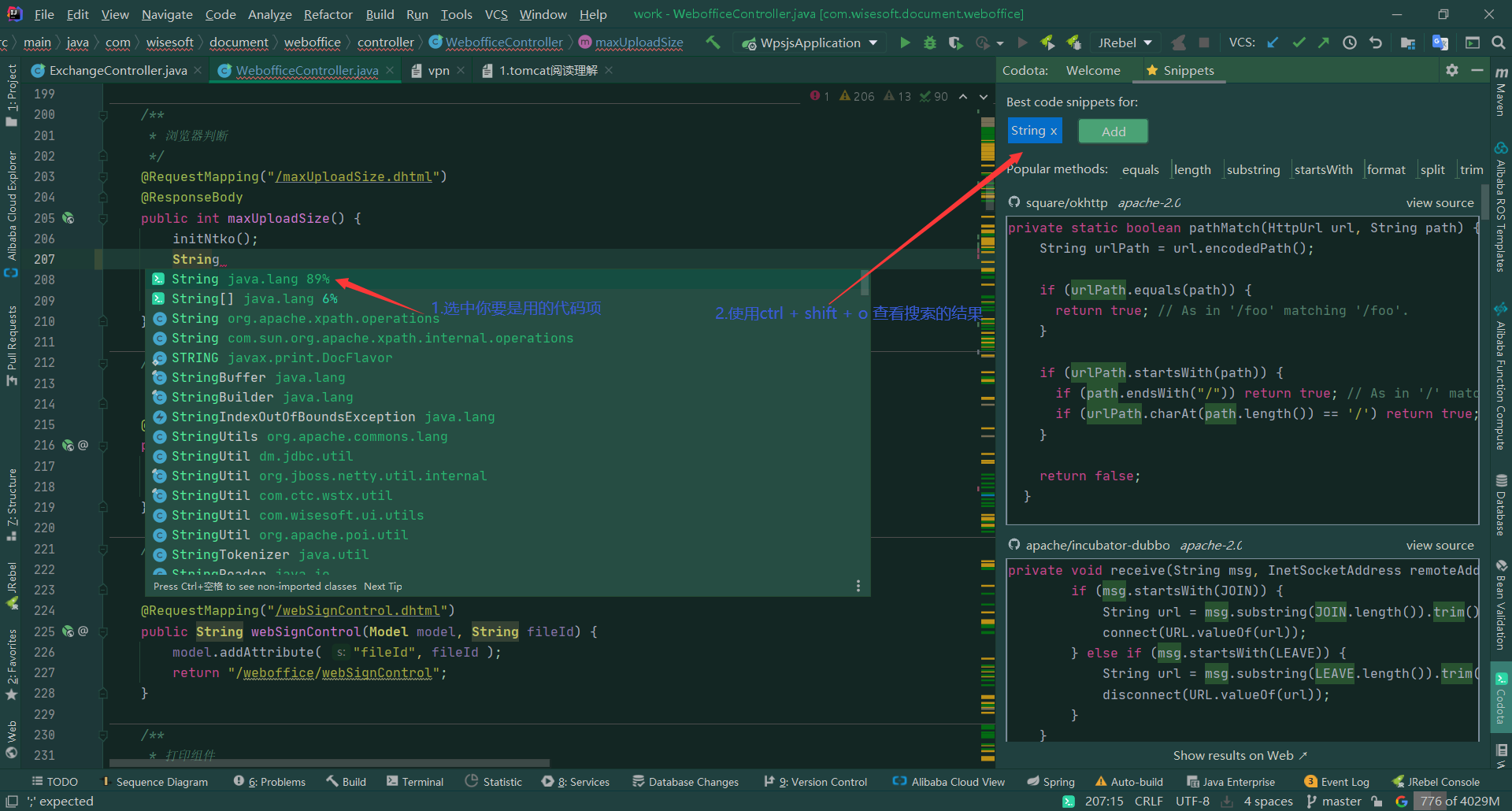
### 2.3 Grep Console插件

可以将控制台打印的日志分颜色显示，可以如下进行设置



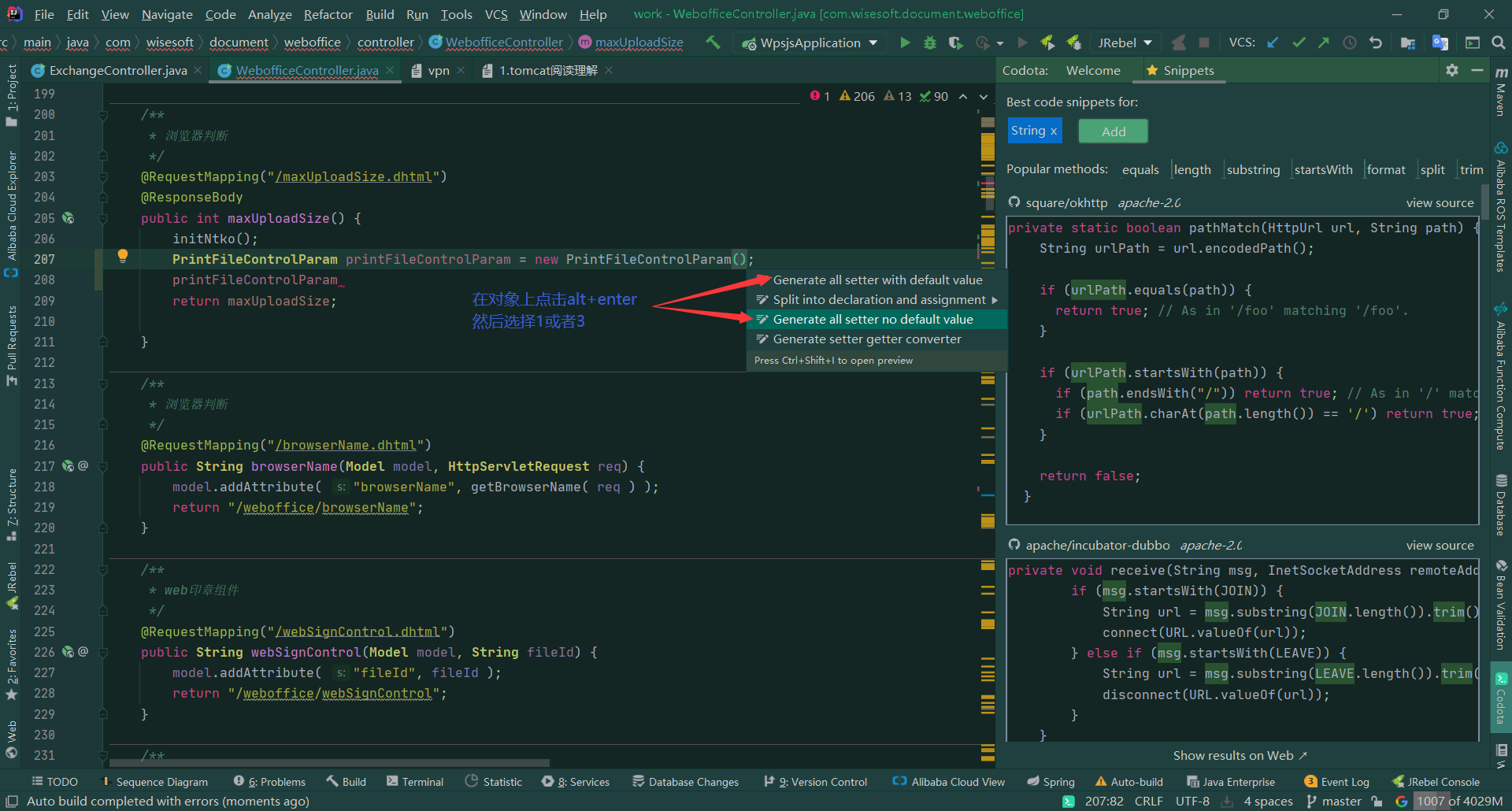
### 2.4 Codota插件

支持代码联想时查看别个大神或者组织是怎么使用该用例或方法。如下：



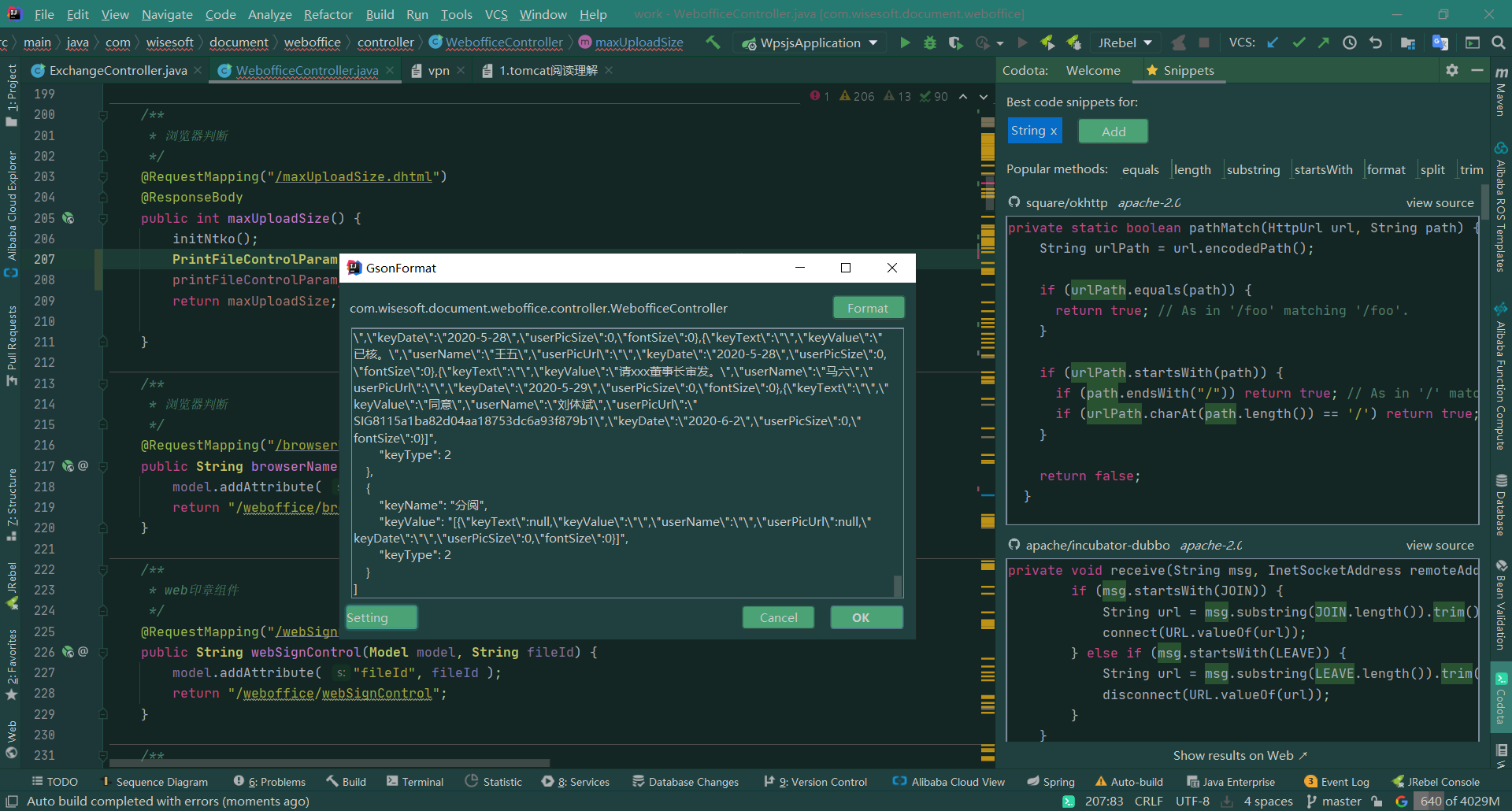
### 2.5 GenerateAllSetter插件

一键生成一个对象的所有set方法



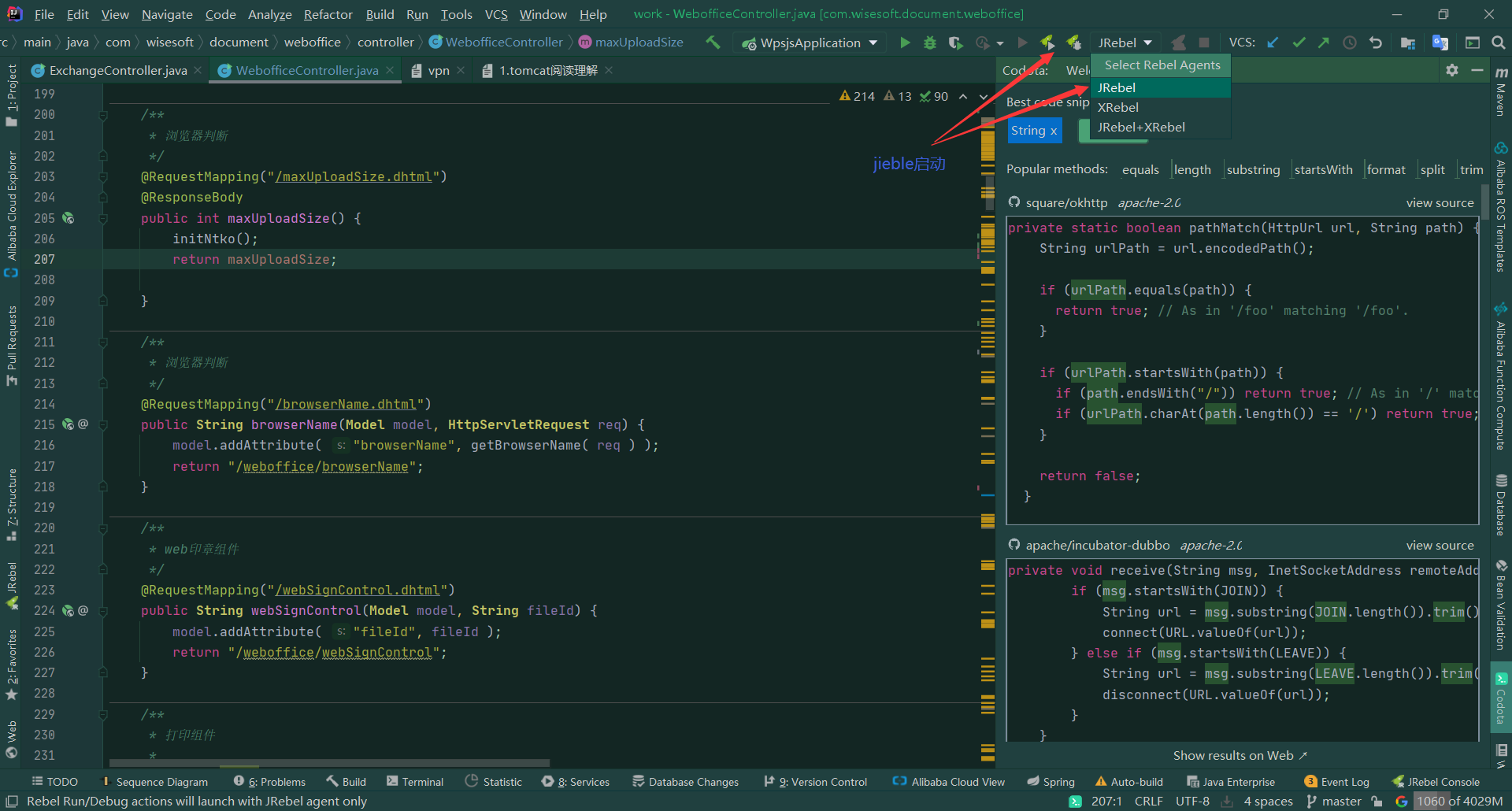
### 2.6 GsonFormat插件

（https://blog.csdn.net/lazycheerup/article/details/84398200），按**alt+s** 快捷键调起，如图。



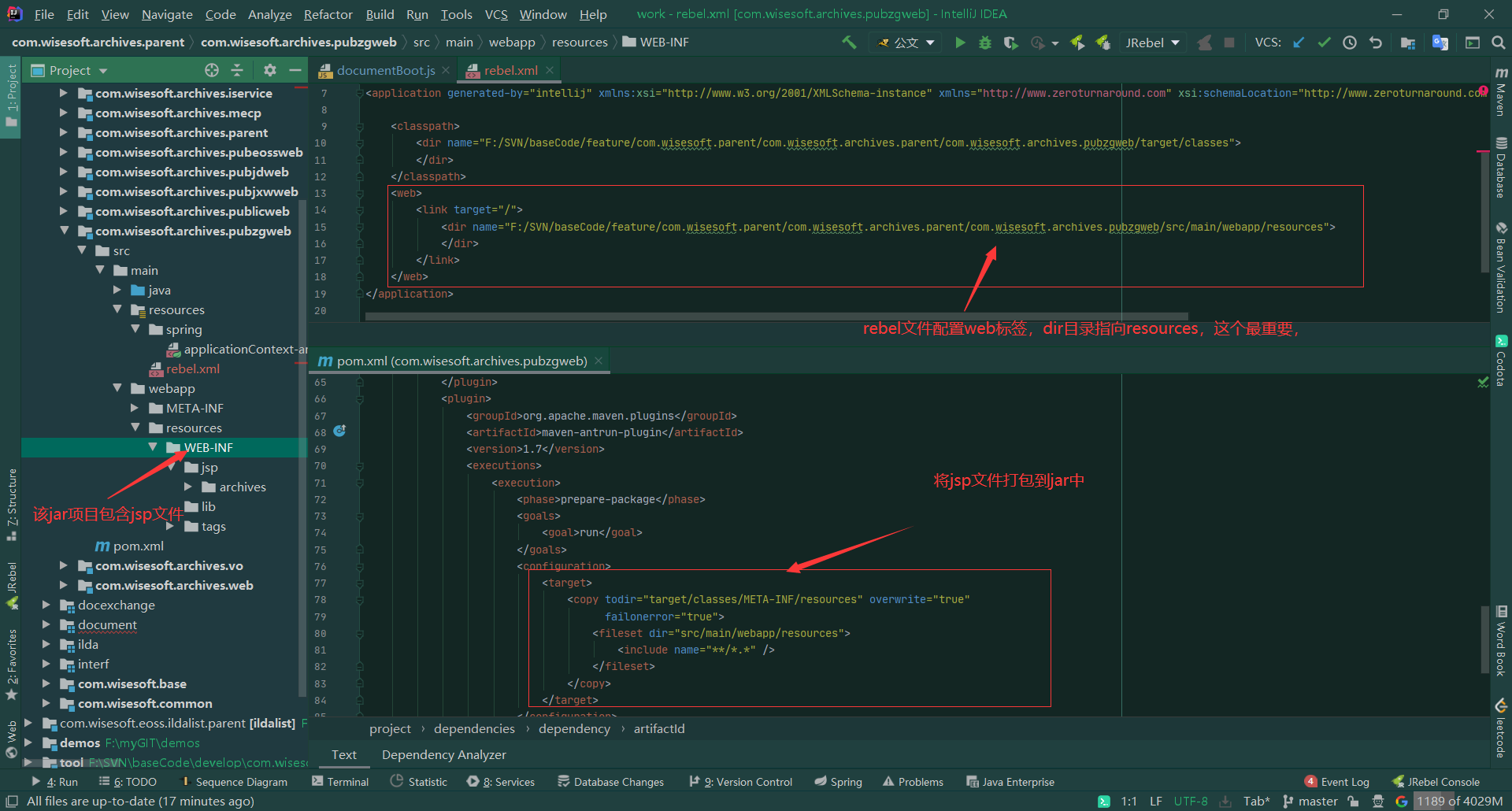
### 2.7 JRebel and XRebel插件

热部署插件。



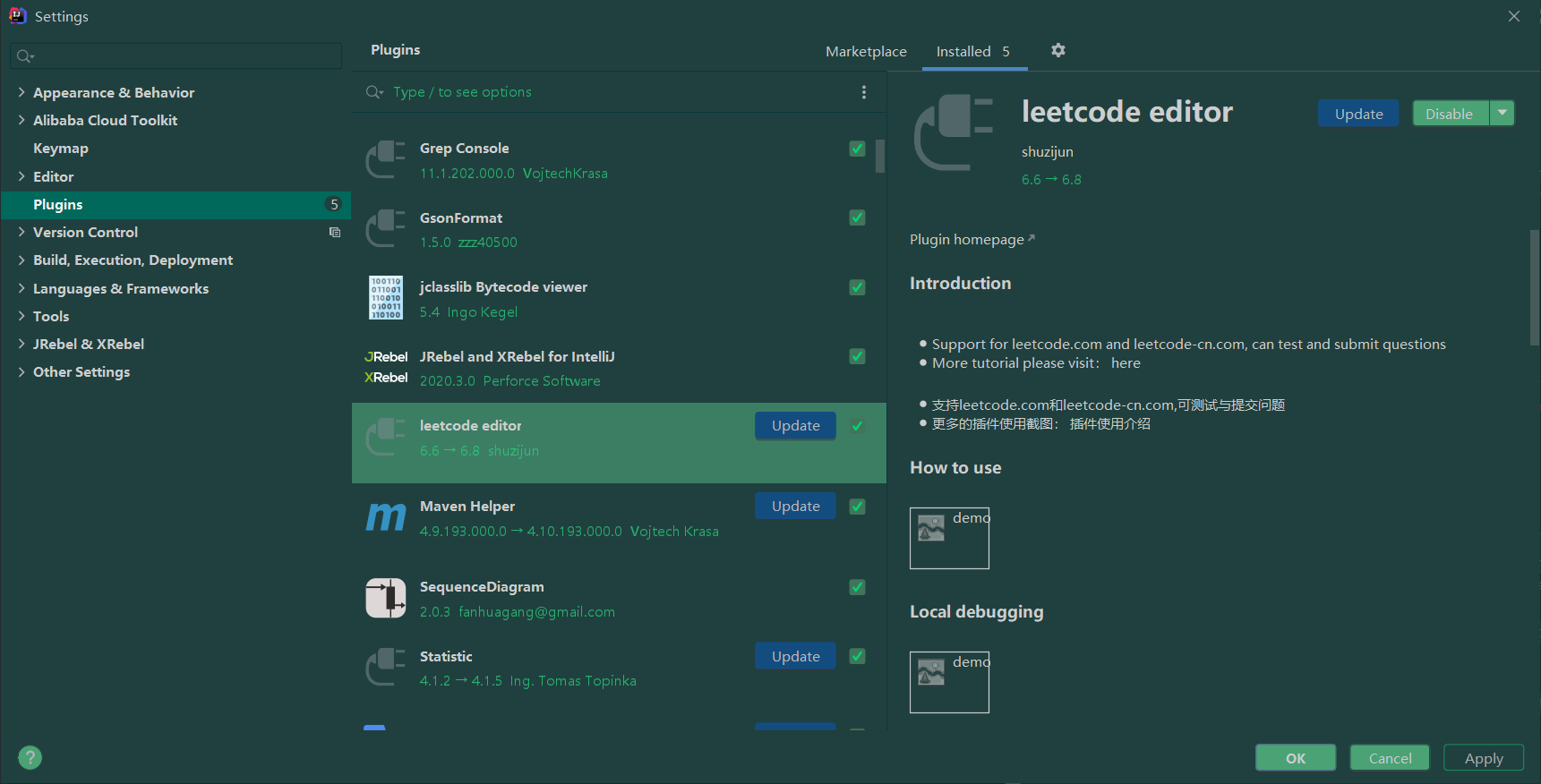
Tomcat启动相关配置: CATALINA\_BASE=E:\下载文件\java产品\tomcat\apache-tomcat-7.0.76-windows-x64\apache-tomcat-7.0.76

热部署写在jar包中的jsp , js 等代码时，需要在在pom文件中配置相应的打包代码。然后再rebel.xml部署jsp等页面的监听，配置如下图：



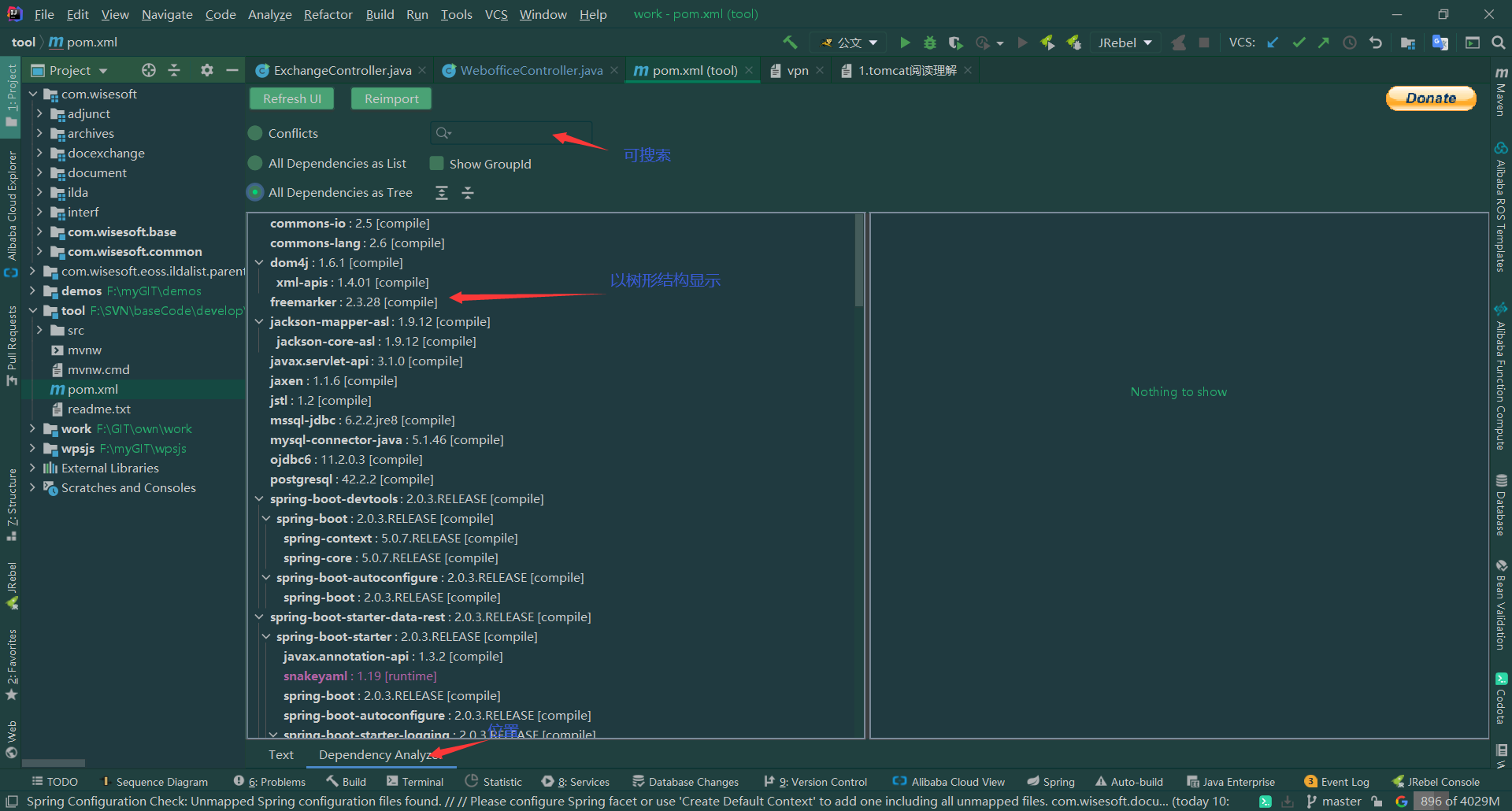
### 2.8 leetcode editor插件

一个连接leetcode官网的插件。



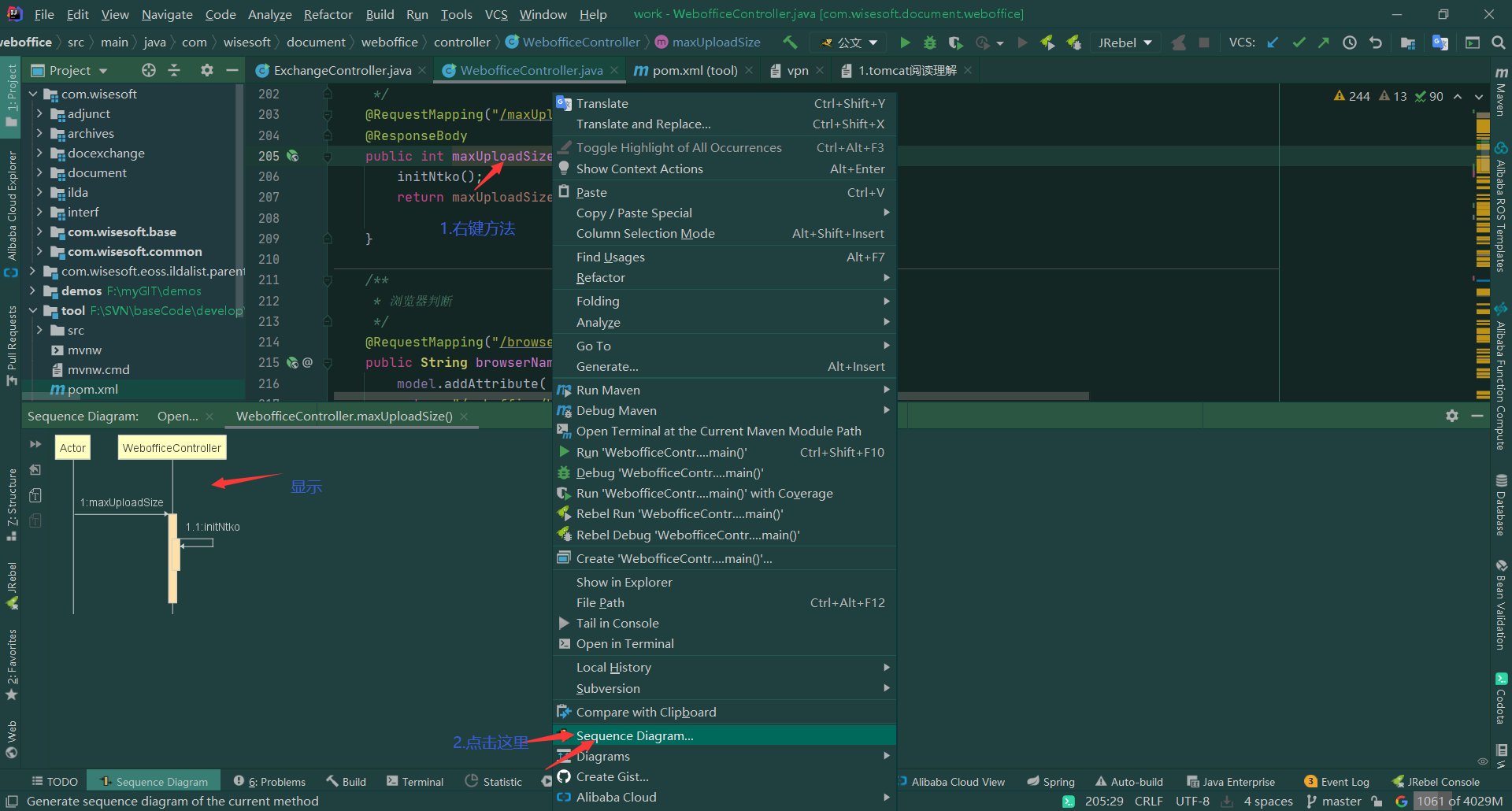
### 2.9 maven helper插件

一个帮组你显示maven依赖树的插件。点击pom文件查看



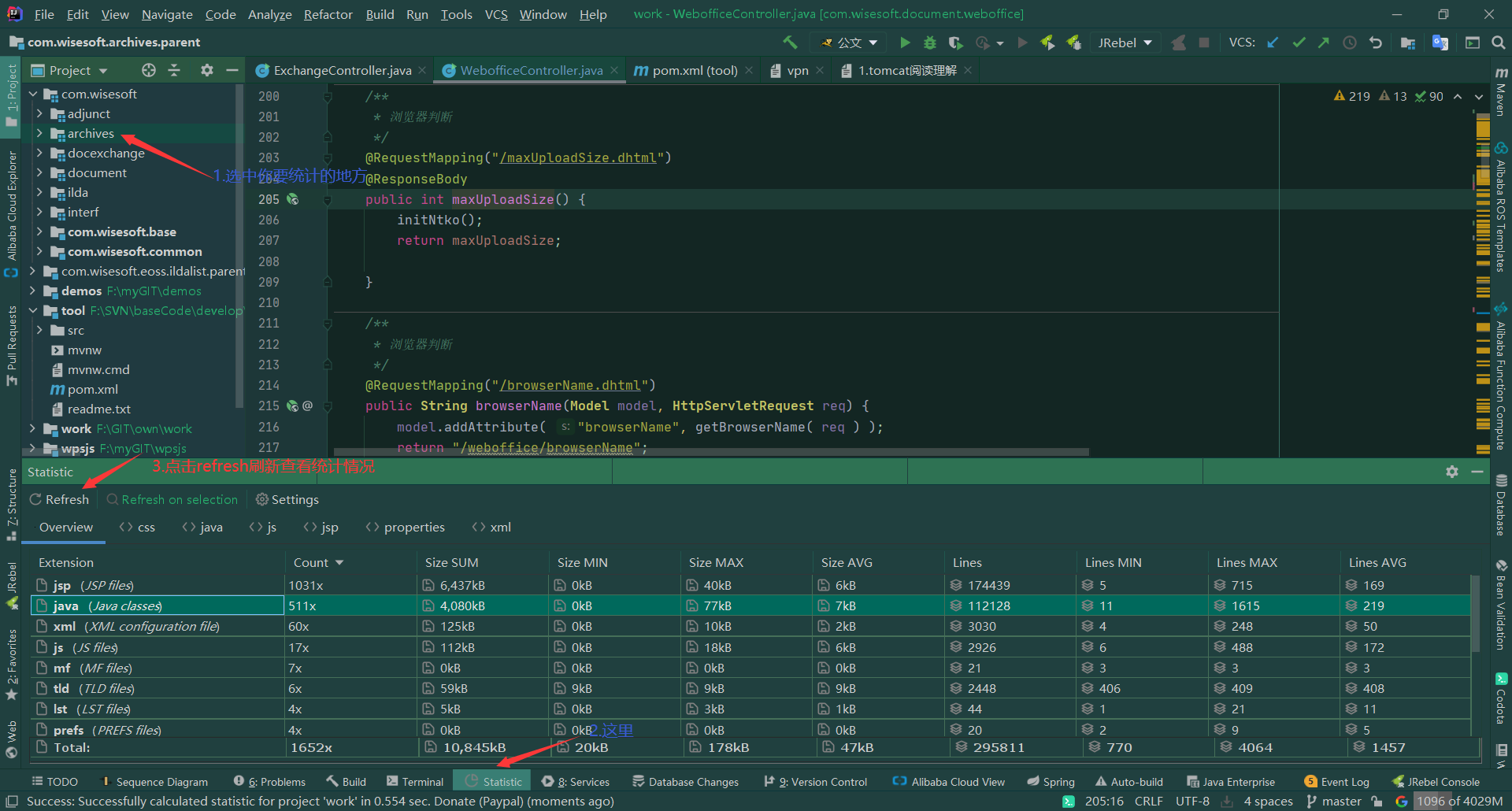
### 2.10 SequenceDiagram插件

查看方法调用时序图

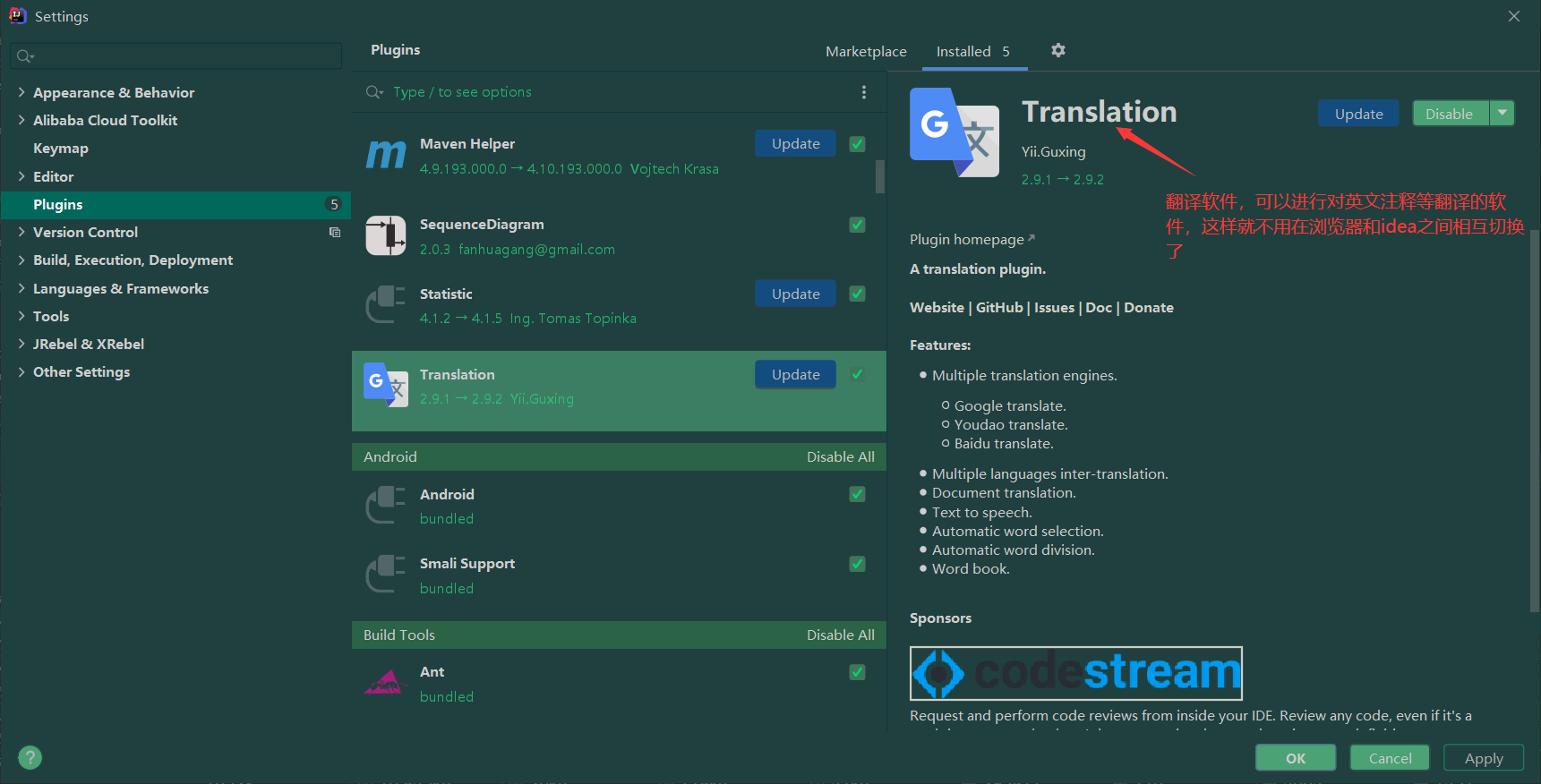


### 2.11 Statistic插件

查看代码大小统计的一个统计查看



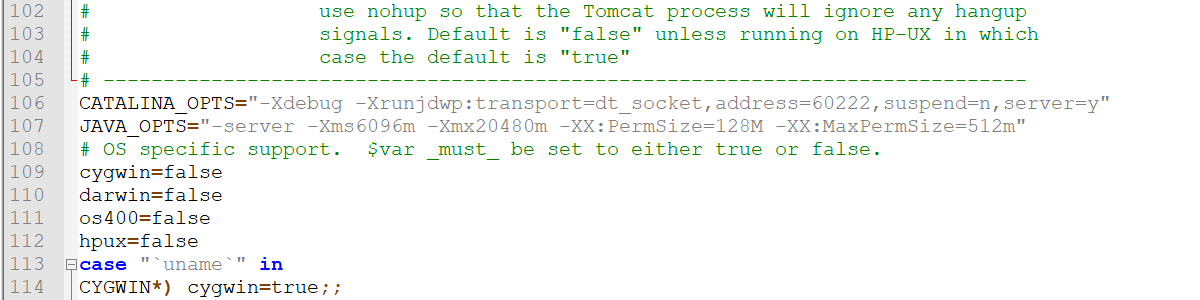
### 2.12 Translation 插件



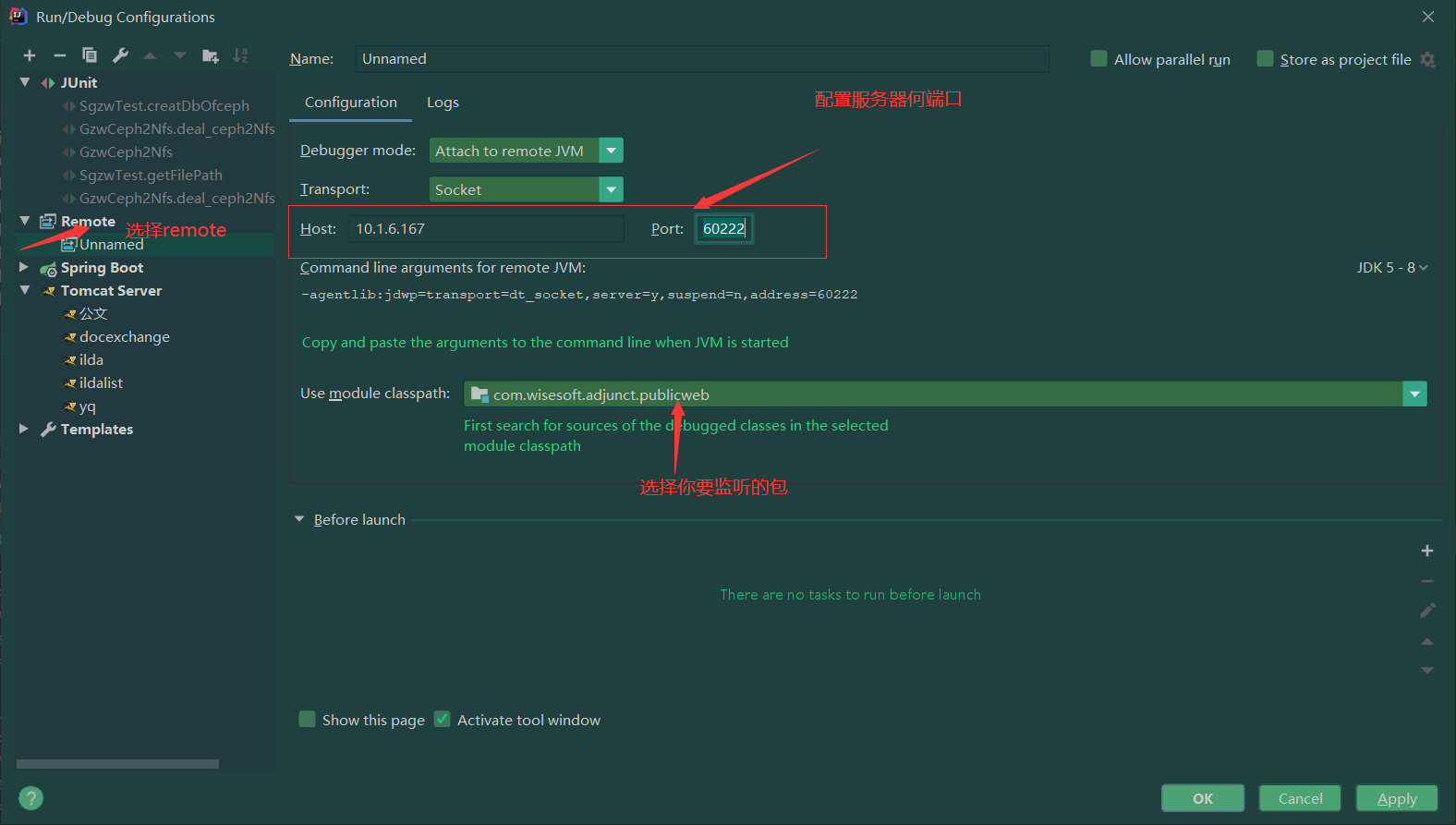
## 3.idea远程调试（remote debugger）

1.首先在服务器端的tomcat的catalina.sh的开始位置配置可以远程调试的配置。如下：**CATALINA\_OPTS="-Xdebug Xrunjdwp:transport=dt\_socket,address=60222,suspend=n,server=y"**

**其他配置：JAVA\_OPTS="-server -Xms6096m -Xmx20480m -XX:PermSize=128M -XX:MaxPermSize=512m"（配置jvm的内存分配配置。）**



2.然后在idea中进行debug的远程监听配置

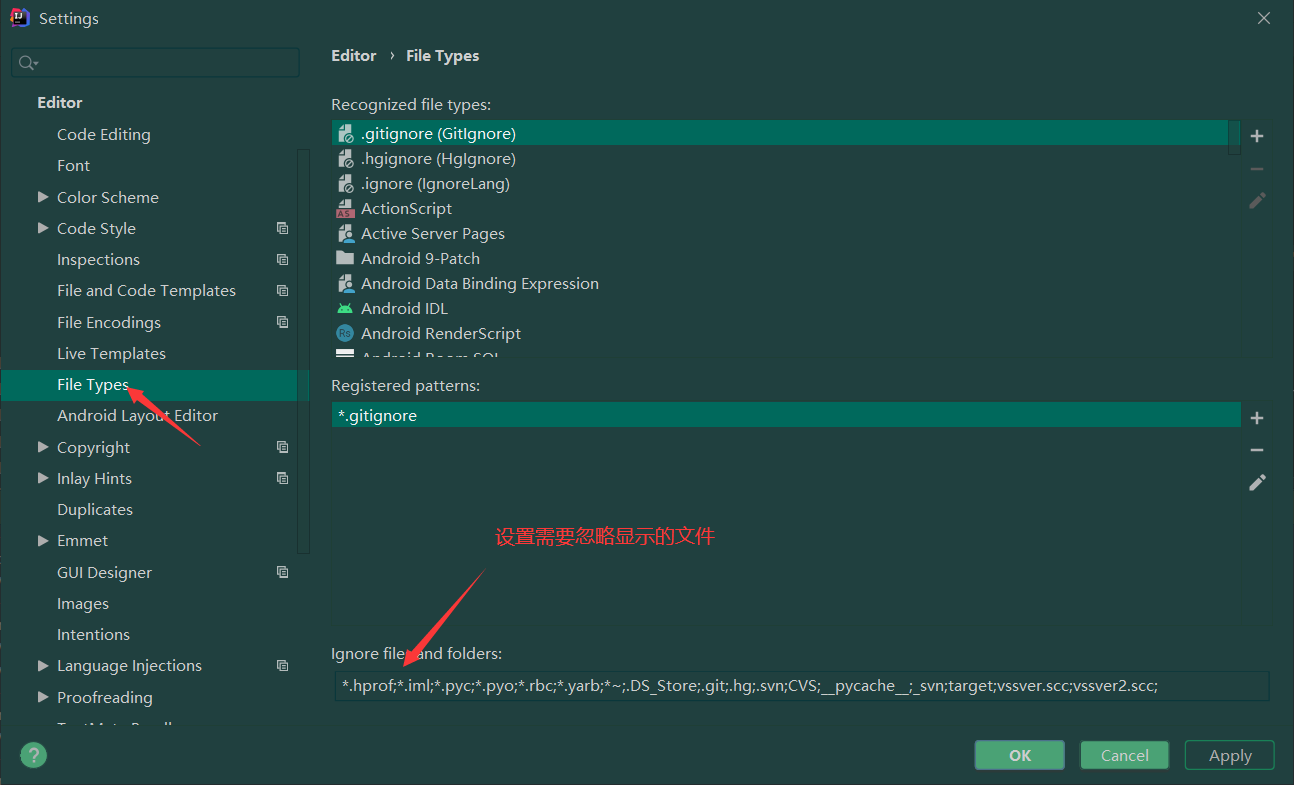


## 4.idea部分设置说明

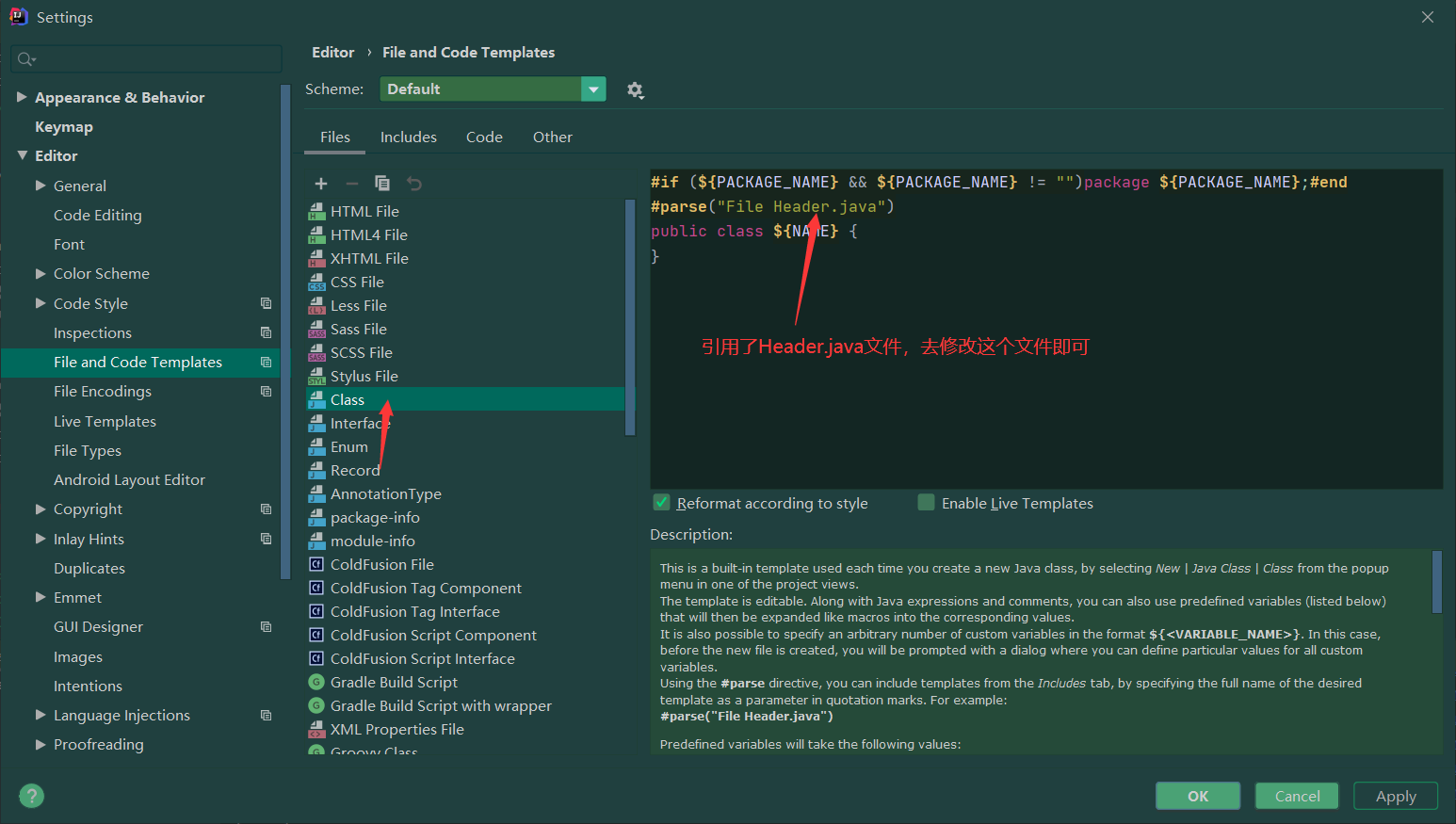
### 4.1 设置project导航忽略文件

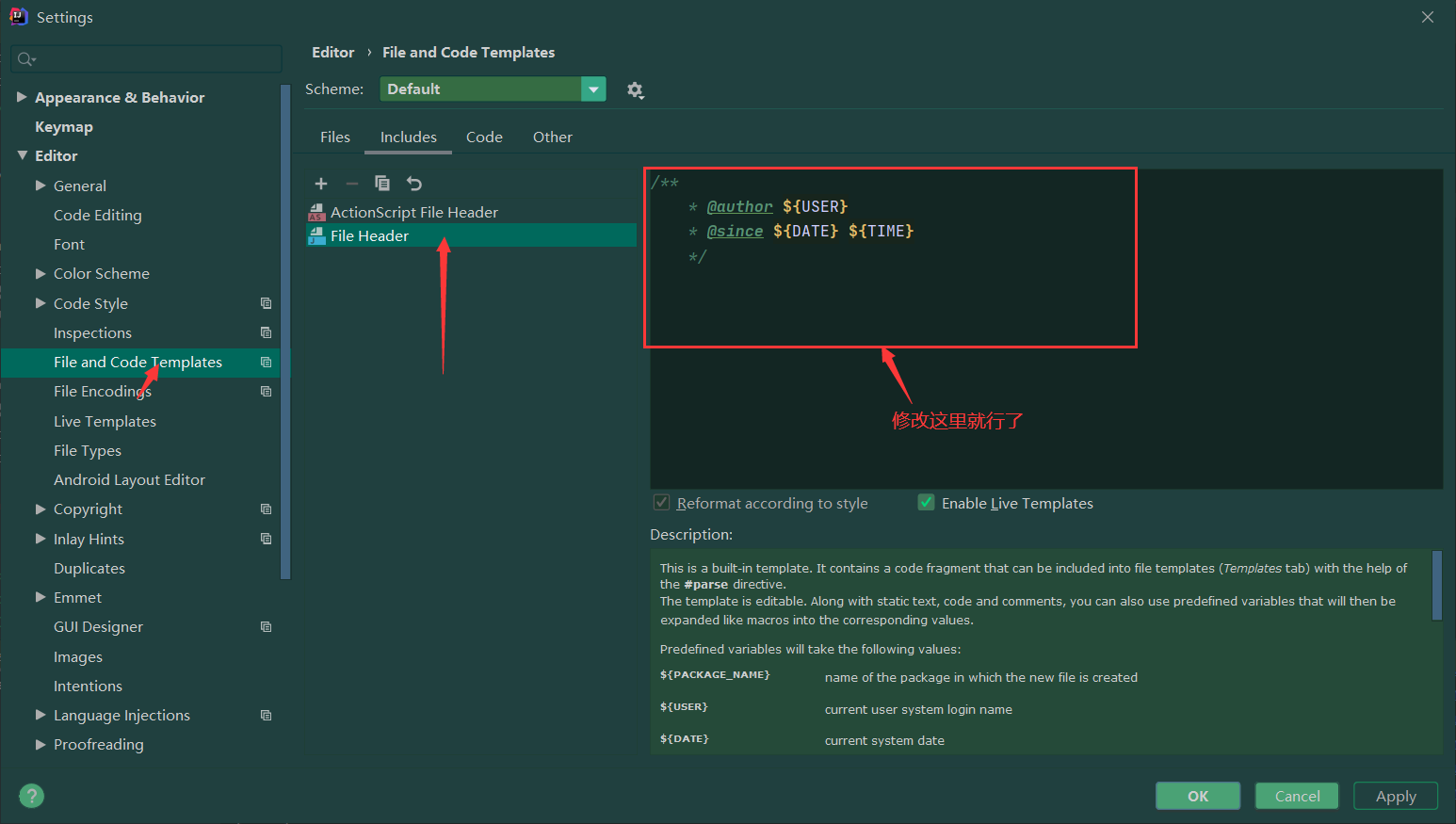
有时候导入项目后project显示文件过多，部分不需要得也显示了出来。这时候可以进行设置把文件 。

**\*.hprof;\*.iml;\*.pyc;\*.pyo;\*.rbc;\*.yarb;\*~;.DS\_Store;.git;.hg;.svn;CVS;\_\_pycache\_\_;\_svn;target;vssver.scc;vssver2.scc;**



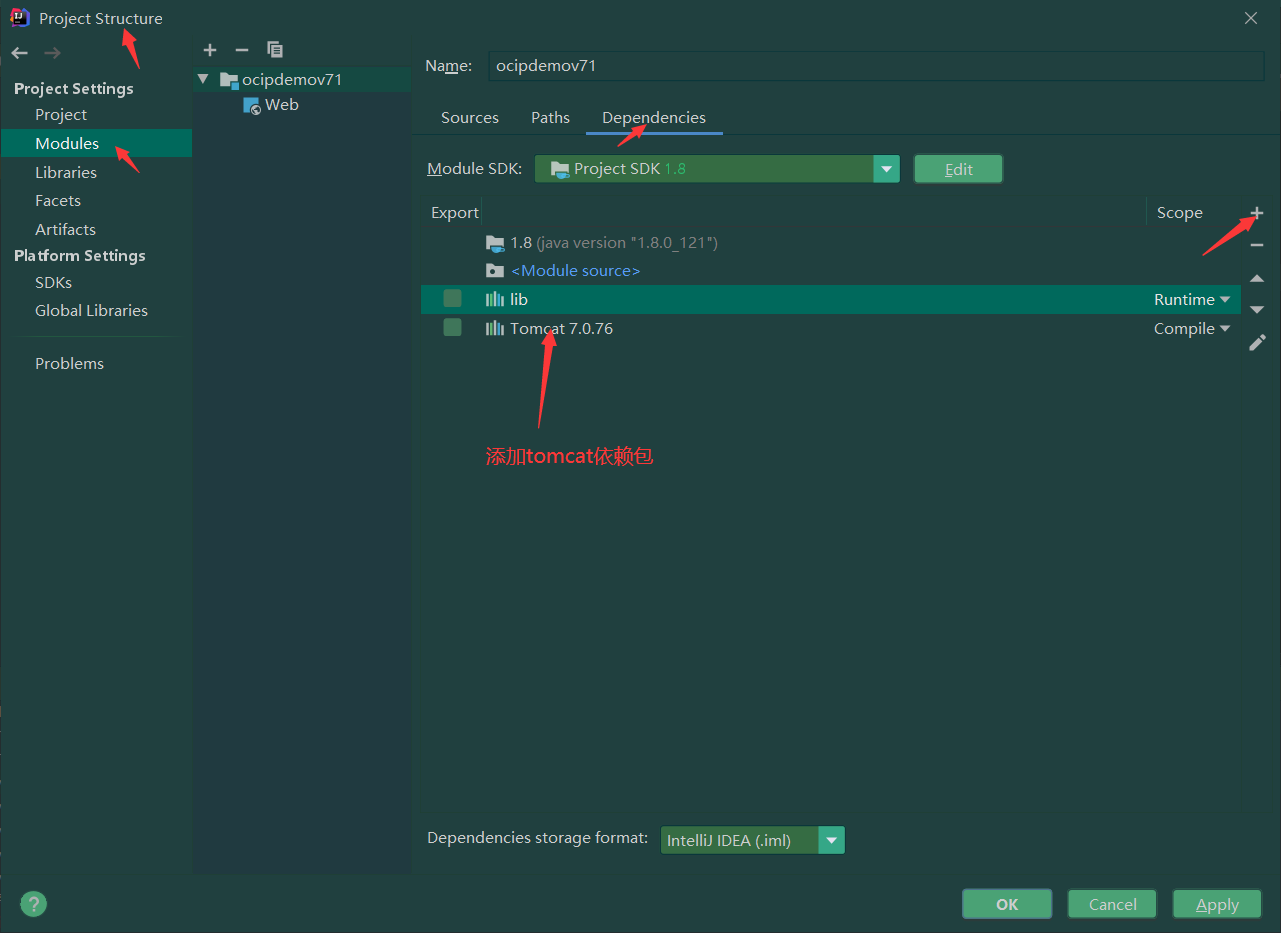
### 4.2 设置类注释模板





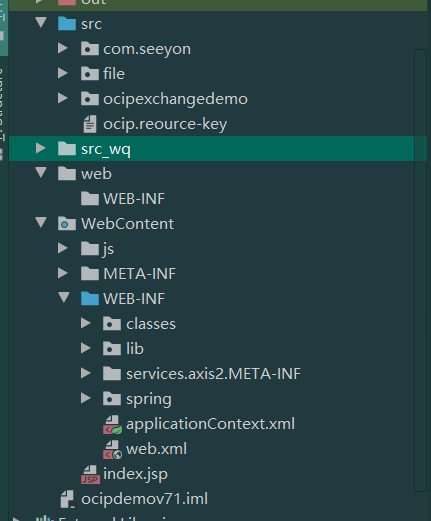
## 5.idea跑传统的web工程

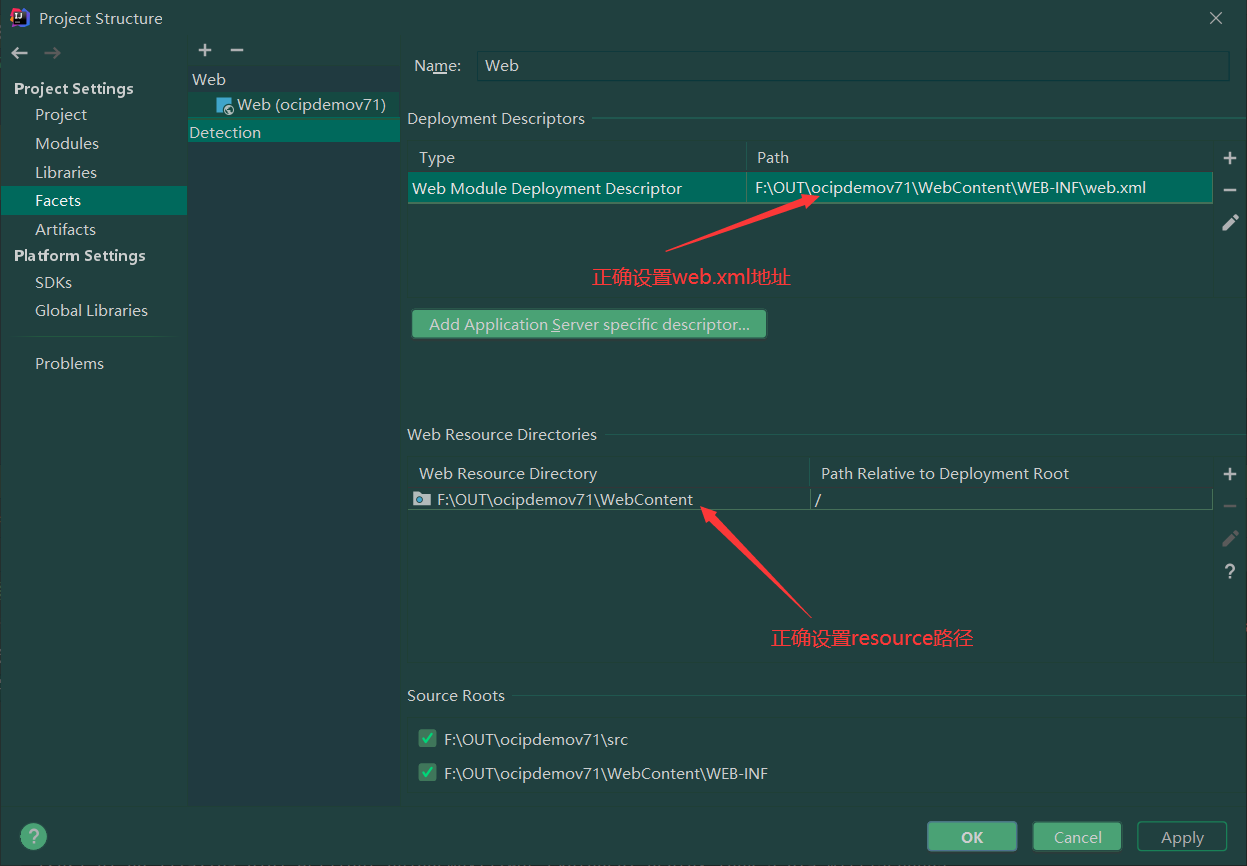
### 5.1 工程缺失servlet.api包

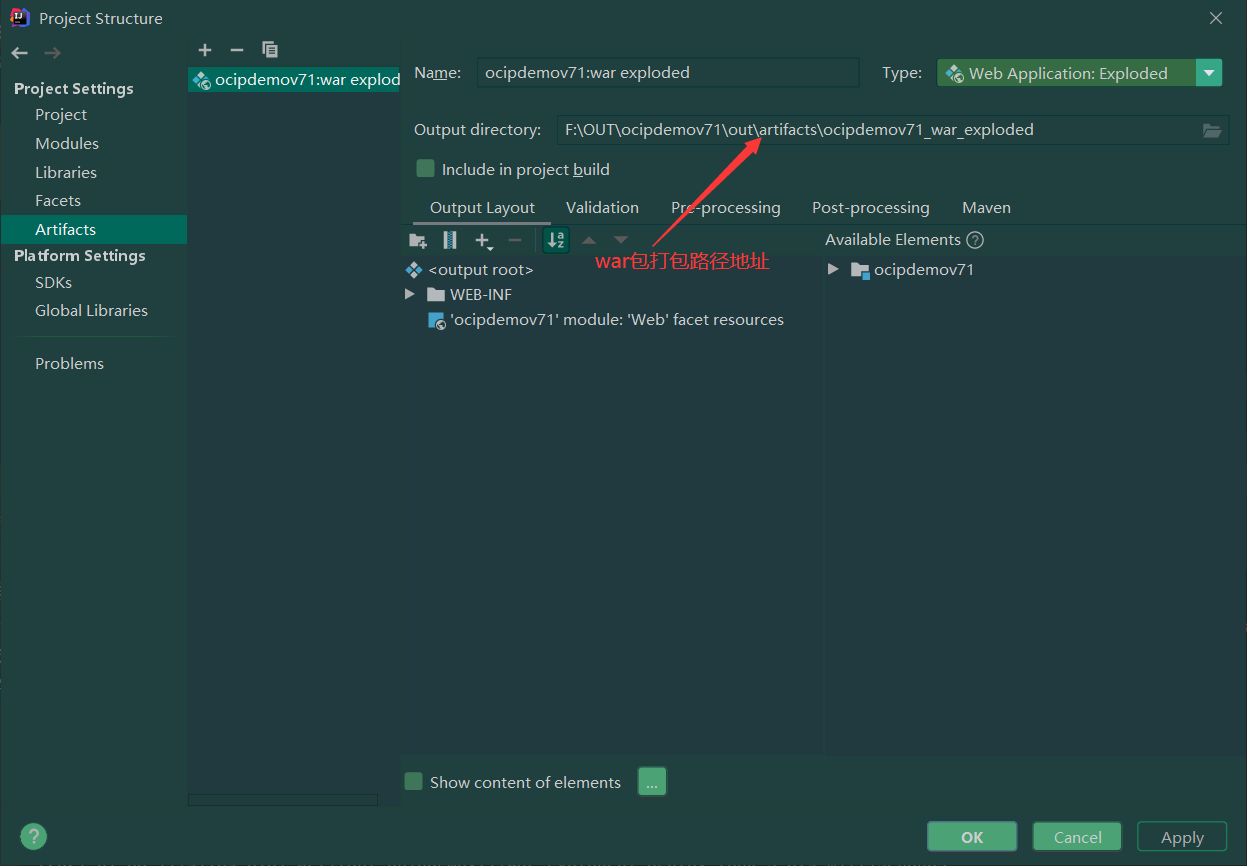


### 5.2 idea的tomcat跑传统web项目的配置方式

公文目录结构：



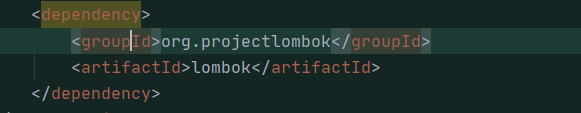




## 6.idea Lombok的日志

1.首先下载lombok插件。

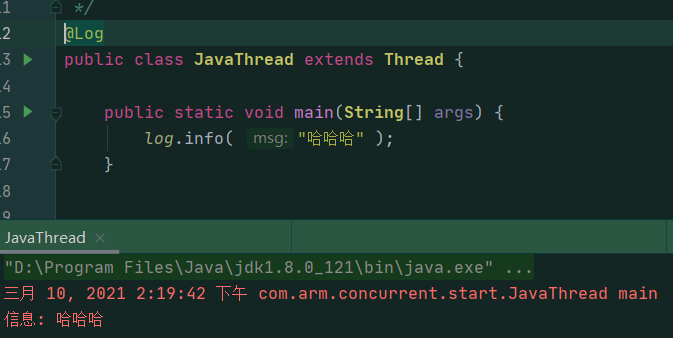
2.项目中导入lombok的jar包



3. 在类上使用想对应的注解

@Log

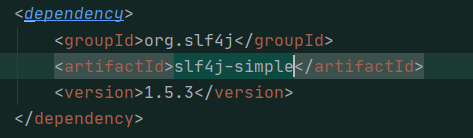
不需要导包，使用的是java默认的日志输出框架：java.util.logging.Logger



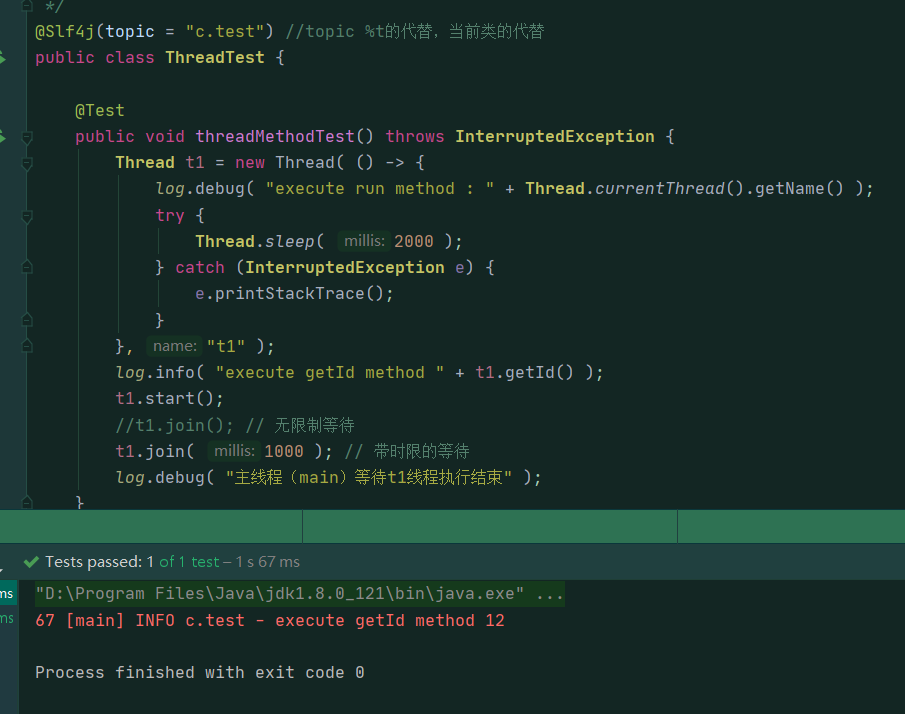
@Slf4j

这是一个框架，其可以与 slf4j-simple log4j等想结合使用

1. 与slf4j-simple想结合 1.导入pom包

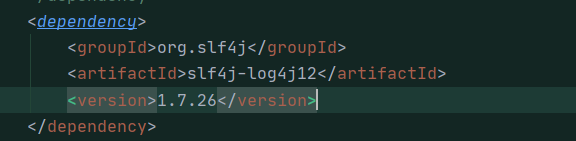


2.代码如下，但是默认的只能输出info级别以上的日志，可以通过配置文件进行修改

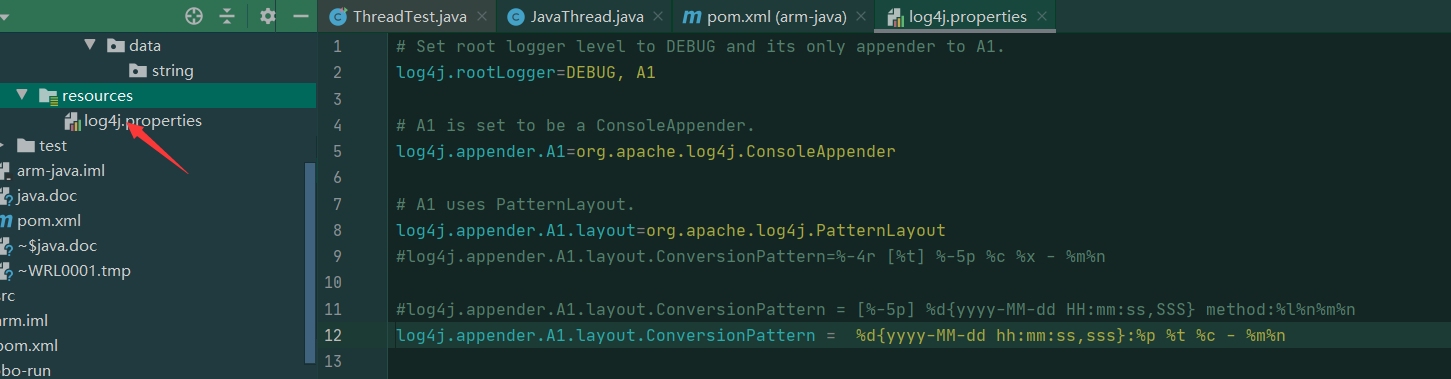


2与log4j的结合 注意事项：必须在项目目录下配置配置文件，不配置无法运行

导入包：

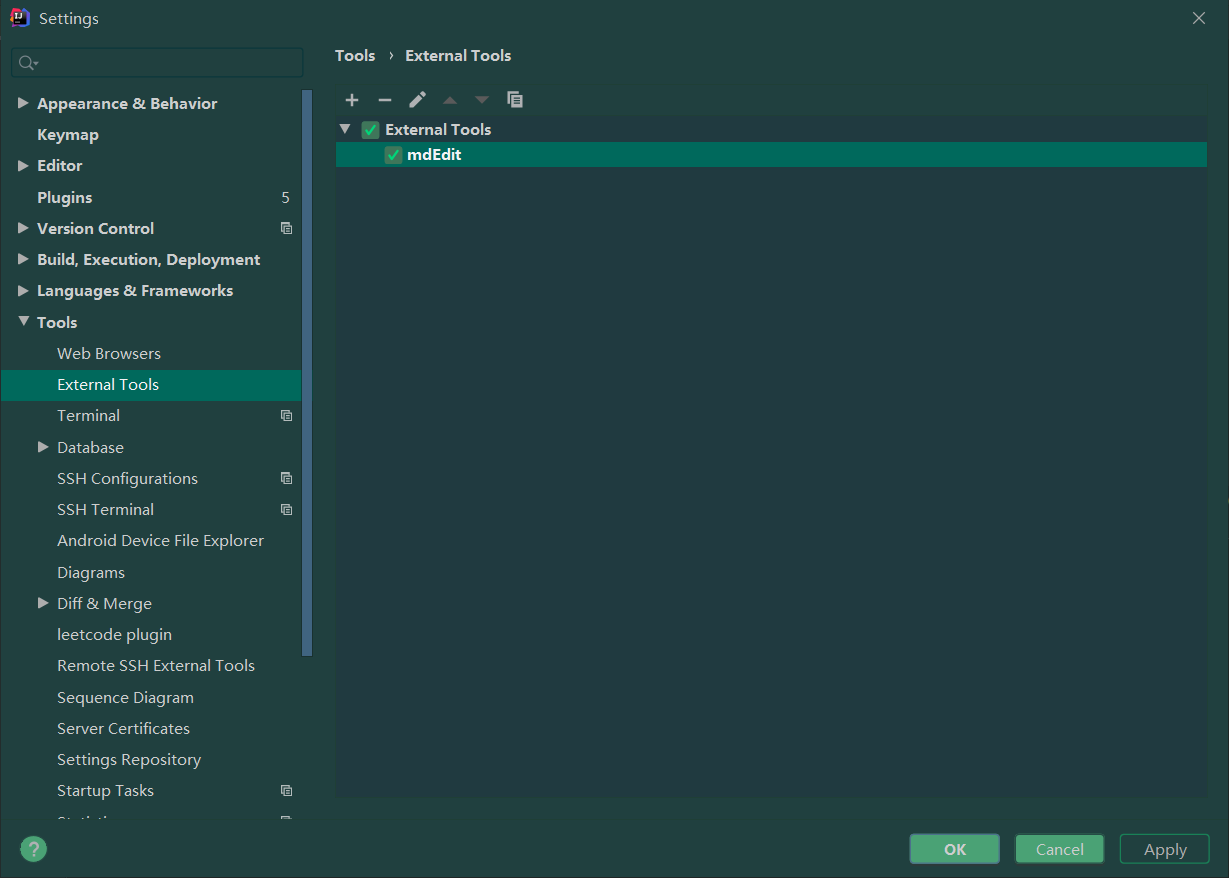


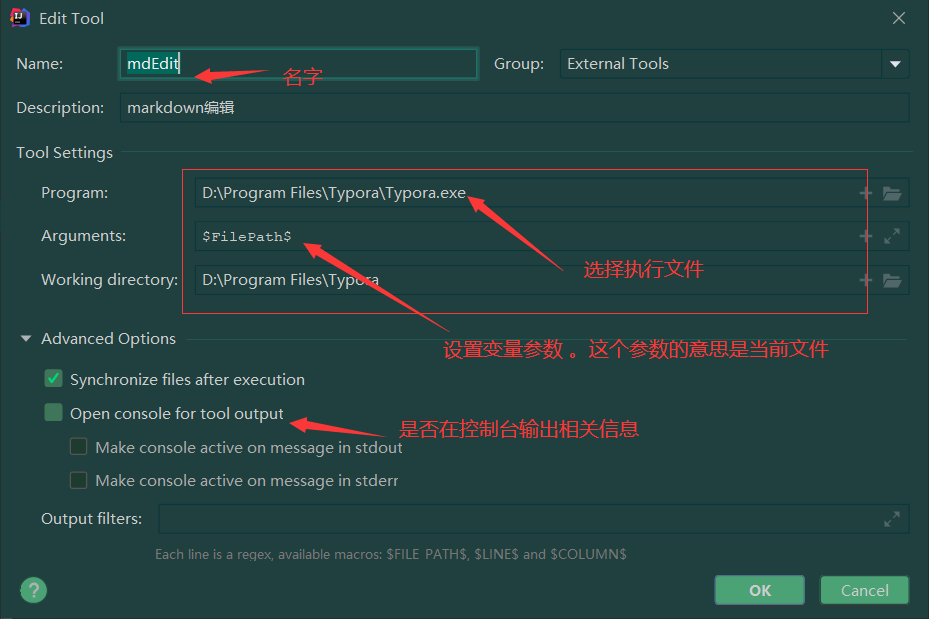
配置文件：



## 7.idea 扩展命令

1.设置扩展命令





2.设置快捷键：设置快捷键后就可以在idea执行执行命令用对应的程序打开文件了。

