**阿里JAVA编码规范使用文档**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **更改描述** | **作者** |
| **2019/03/05** | **V1.0** | 文档初创 | **李加利** |

目录

[1. 引言](#_Toc1193_WPSOffice_Level1) [2](#_Toc1193_WPSOffice_Level1)

[1.1. 目的](#_Toc14036_WPSOffice_Level2) [2](#_Toc14036_WPSOffice_Level2)

[1.2. 背景](#_Toc5923_WPSOffice_Level2) [2](#_Toc5923_WPSOffice_Level2)

[1.3. 参考资料](#_Toc28503_WPSOffice_Level2) [2](#_Toc28503_WPSOffice_Level2)

[2. 阿里巴巴JAVA开发手册介绍](#_Toc14036_WPSOffice_Level1) [2](#_Toc14036_WPSOffice_Level1)

[2.1. 说明](#_Toc838_WPSOffice_Level2) [2](#_Toc838_WPSOffice_Level2)

[2.2. 下载](#_Toc6659_WPSOffice_Level2) [3](#_Toc6659_WPSOffice_Level2)

[3. 插件介绍](#_Toc5923_WPSOffice_Level1) [3](#_Toc5923_WPSOffice_Level1)

[4. 阿里代码格式设置](#_Toc28503_WPSOffice_Level1) [3](#_Toc28503_WPSOffice_Level1)

[4.1. eclipse代码格式规范](#_Toc9628_WPSOffice_Level2) [3](#_Toc9628_WPSOffice_Level2)

[4.1.1. 下载](#_Toc14036_WPSOffice_Level3) [3](#_Toc14036_WPSOffice_Level3)

[4.1.2. 注释模板配置](#_Toc5923_WPSOffice_Level3) [4](#_Toc5923_WPSOffice_Level3)

[4.1.3. 新建代码格式样式](#_Toc28503_WPSOffice_Level3) [6](#_Toc28503_WPSOffice_Level3)

[4.1.4. 导入代码格式化样式](#_Toc838_WPSOffice_Level3) [6](#_Toc838_WPSOffice_Level3)

[4.1.5. eclipse设置保存时自动格式化（可选）](#_Toc6659_WPSOffice_Level3) [7](#_Toc6659_WPSOffice_Level3)

[4.2. Idea代码格式规范](#_Toc32020_WPSOffice_Level2) [8](#_Toc32020_WPSOffice_Level2)

[4.2.1. 下载](#_Toc9628_WPSOffice_Level3) [8](#_Toc9628_WPSOffice_Level3)

[4.2.2. eclipseFormatter插件安装](#_Toc32020_WPSOffice_Level3) [9](#_Toc32020_WPSOffice_Level3)

[4.2.3. Save Actions插件安装](#_Toc2769_WPSOffice_Level3) [10](#_Toc2769_WPSOffice_Level3)

[4.2.4. 注释模板配置](#_Toc23646_WPSOffice_Level3) [12](#_Toc23646_WPSOffice_Level3)

[5 设置模板的应用场景](#_Toc838_WPSOffice_Level1) [14](#_Toc838_WPSOffice_Level1)

[（6）设置参数的获取方式](#_Toc2769_WPSOffice_Level2) [15](#_Toc2769_WPSOffice_Level2)

[（7）效果图](#_Toc23646_WPSOffice_Level2) [15](#_Toc23646_WPSOffice_Level2)

[4.2.5. 新建代码格式样式](#_Toc4570_WPSOffice_Level3) [16](#_Toc4570_WPSOffice_Level3)

[4.2.6. 导入代码格式化样式（导入eclipse-codestyle.xml）](#_Toc25513_WPSOffice_Level3) [16](#_Toc25513_WPSOffice_Level3)

[4.2.7. Idea设置保存时自动格式化（可选）](#_Toc1380_WPSOffice_Level3) [17](#_Toc1380_WPSOffice_Level3)

[5.1. eclipse编码规约](#_Toc4570_WPSOffice_Level2) [17](#_Toc4570_WPSOffice_Level2)

[5.1.1. 插件下载](#_Toc20717_WPSOffice_Level3) [17](#_Toc20717_WPSOffice_Level3)

[5.1.2. 插件安装](#_Toc14742_WPSOffice_Level3) [18](#_Toc14742_WPSOffice_Level3)

[5.1.3. 编写测试文件](#_Toc17786_WPSOffice_Level3) [18](#_Toc17786_WPSOffice_Level3)

[5.1.4. 点击进行阿里编码规约扫描](#_Toc6438_WPSOffice_Level3) [18](#_Toc6438_WPSOffice_Level3)

[5.2. Idea编码规约](#_Toc25513_WPSOffice_Level2) [19](#_Toc25513_WPSOffice_Level2)

[5.2.1. 插件下载](#_Toc7500_WPSOffice_Level3) [19](#_Toc7500_WPSOffice_Level3)

[5.2.2. 插件安装](#_Toc9165_WPSOffice_Level3) [20](#_Toc9165_WPSOffice_Level3)

[5.2.3. 插件使用](#_Toc26143_WPSOffice_Level3) [22](#_Toc26143_WPSOffice_Level3)

[6.1. 简介](#_Toc1380_WPSOffice_Level2) [25](#_Toc1380_WPSOffice_Level2)

[6.2. Eclipse插件安装](#_Toc20717_WPSOffice_Level2) [25](#_Toc20717_WPSOffice_Level2)

[6.2.1. 在线安装](#_Toc12699_WPSOffice_Level3) [25](#_Toc12699_WPSOffice_Level3)

[6.2.2. 离线安装](#_Toc7166_WPSOffice_Level3) [27](#_Toc7166_WPSOffice_Level3)

[6.2.3. findBugs配置](#_Toc10690_WPSOffice_Level3) [27](#_Toc10690_WPSOffice_Level3)

[6.2.4. findBugs使用](#_Toc16221_WPSOffice_Level3) [31](#_Toc16221_WPSOffice_Level3)

[6.3. Idea插件安装](#_Toc14742_WPSOffice_Level2) [33](#_Toc14742_WPSOffice_Level2)

[6.3.1. 下载](#_Toc4297_WPSOffice_Level3) [33](#_Toc4297_WPSOffice_Level3)

[6.3.2. findBugs安装](#_Toc14595_WPSOffice_Level3) [34](#_Toc14595_WPSOffice_Level3)

[6.3.3. findBugs配置](#_Toc12614_WPSOffice_Level3) [35](#_Toc12614_WPSOffice_Level3)

[6.3.4. findBugs使用](#_Toc25172_WPSOffice_Level3) [35](#_Toc25172_WPSOffice_Level3)

[7.1. 阿里编码规约](#_Toc17786_WPSOffice_Level2) [36](#_Toc17786_WPSOffice_Level2)

[7.2. findBugs扫描结果](#_Toc6438_WPSOffice_Level2) [37](#_Toc6438_WPSOffice_Level2)

# 引言

## 目的

编写此文档是为了说明阿里p3c插件的安装以及使用方法，本文档的阅读对象是系统研发人员。

## 背景

目前公司在系统架构设计完成之后,编码工作开始之前,尚未有属于自己的编码规范文档,所有的编码工作人员都必须遵守其中的规范,避免规范不统一带来的不必要的沟通问题,针对该问题编写该使用文档来统一规范标准。

**养成良好的编程规范的作用(为何学习并养成编程风格)**

1. 好的编码规范可以尽可能的减少一个软件的维护成本，并且几乎没有任何一个软件，其整个生命周期中，均由最初的开发人员来维护；

2、好的编码规范可以改善软件的可读性，可以让开发人员尽快而彻底地理解新的代码；

3、好的编码规范可以最大限度的提高团队开发的合作效率；

4、长期的规范性编码还可以让开发人员养成好的编码习惯，甚至锻炼出更加严谨的思维；

## 参考资料

《阿里巴巴Java开发手册》

# 阿里巴巴JAVA开发手册介绍

## 说明

在进入正题介绍这款插件之前，首先来谈一下《阿里巴巴Java开发手册》，2017年年初，首次公开的阿里官方Java代码规范标准手册可以说是引起了全民(IT界)代码规范的热潮，相信这个手册很多行业内的朋友都比较熟悉，毕竟有"大厂光环"，想低调都难，这个手册的愿景是:

**"统一规范标准将有助于提高行业编码规范化水平，帮助行业人员提高开发质量和效率、大大降低代码维护成本"。**

## 下载



# 插件介绍

代码规范检查插件p3c，是根据**《阿里巴巴Java开发手册》**转化而成的自动化插件。

（高级黑：P-3C“Orion”，反潜巡逻机，阿里大概取p3c先进，监测，发现潜在问题的意思）

该插件由阿里巴巴 P3C 项目组研发。P3C 是世界知名的反潜机，专门对付水下潜水艇，寓意是扫描出所有潜在的代码隐患。



代码已经开源，GitHub：**<https://github.com/alibaba/p3c>**  
阿里介绍文章：**[https://mp.weixin.qq.com/s/IbibsXlWHlM59kfXJqRvZA#rd](https://mp.weixin.qq.com/s/IbibsXlWHlM59kfXJqRvZA" \l "rd)**

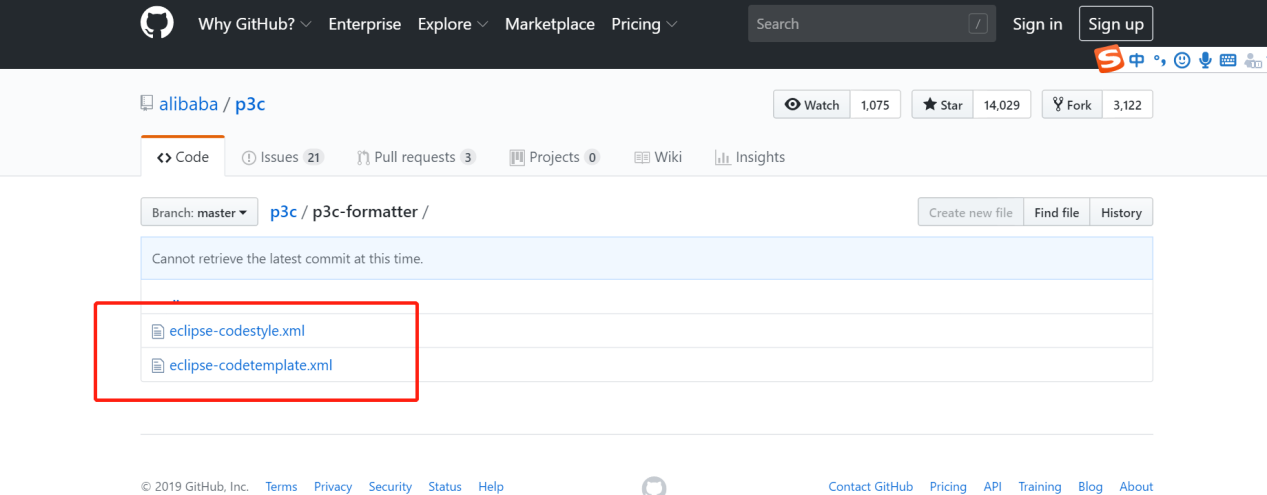
插件支持了IDEA、Eclipse，在扫描代码后，将不符合规约的代码显示出来，甚至在 IDEA 上，我们还基于 Inspection 机制提供了实时检测功能，编写代码的同时也能快速发现问题所在，还实现了批量一键修复的功能。

# 阿里代码格式设置

## eclipse代码格式规范

### 下载

【p3c-formatter】：【[https://github.com/alibaba/p3c/tree/master/p3c-formatter](https://github.com/alibaba/p3c/tree/master/p3c-formatter" \t "http://www.cnblogs.com/lsysy/p/_blank)】



### 注释模板配置

**编辑注释模板的方法：**Window->Preference->Java->Code Style->Code Template** 然后展开**Comments**节点就是所有需设置注释的元素啦。现就每一个元素逐一介绍:**

1. 文件(Files)注释标签

|  |
| --- |
| /\*\*  \* @Title: ${file\_name}  \* @Package ${package\_name}  \* @Description: ${todo}(用一句话描述该文件做什么)  \* @author: ${user}  \* @date: ${date} ${time}  \* @version V1.0  \*/ |

1. 类型(Types)注释标签（类的注释）

|  |
| --- |
| /\*\*  \* @ClassName: ${type\_name}  \* @Description:${todo}(这里用一句话描述这个类的作用)  \* @author: ${user}  \* @date: ${date} ${time}  \* ${tags}  \*/ |

1. 字段(Fields)注释标签

|  |
| --- |
| /\*\*  \* @Fields ${field} : ${todo}(用一句话描述这个变量表示什么)  \*/ |

1. 构造函数标签

|  |
| --- |
| /\*\*  \* @Title: ${enclosing\_type}  \* @Description: ${todo}(这里用一句话描述这个方法的作用)  \* @param: ${tags}  \* @throws  \*/ |

1. 方法(Methods)标签

|  |
| --- |
| /\*\*  \* @Title: ${enclosing\_method}  \* @Description: ${todo}(这里用一句话描述这个方法的作用)  \* @param: ${tags}  \* @return: ${return\_type}  \* @date: ${date} ${time}  \* @throws  \*/ |

1. 覆盖方法(Overriding Methods)标签

|  |
| --- |
| /\*\*  \* <p>Title: ${enclosing\_method}</p>  \* <p>Description:请输入你的描述</p>  \* ${tags}  \* ${see\_to\_overridden}  \*/ |

1. getter方法标签

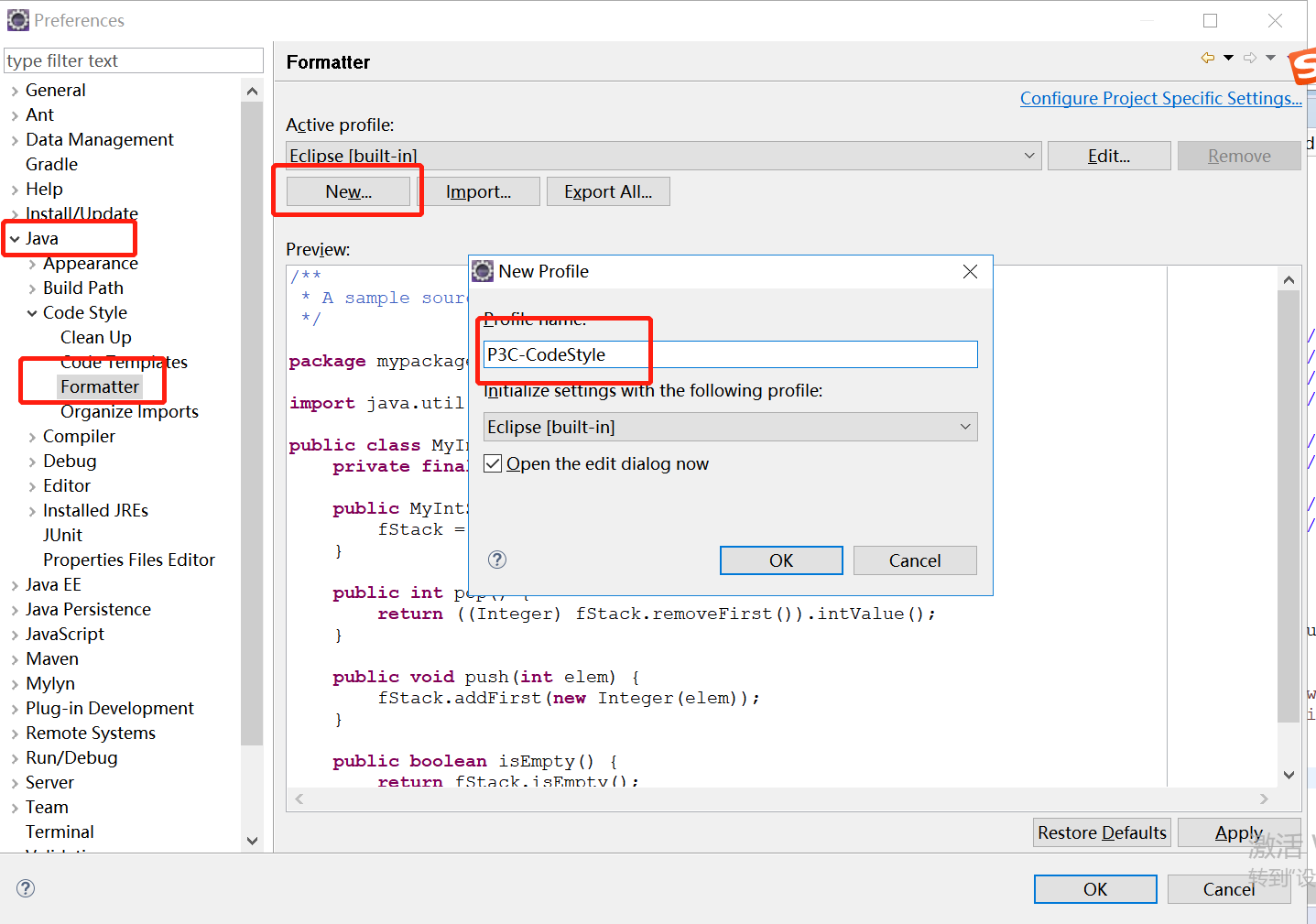
|  |
| --- |
| /\*\*  \* @Title: ${enclosing\_method} <BR>  \* @Description: 请输入你的描述 <BR>  \* @return: ${field\_type} <BR>  \*/ |

1. setter方法标签

|  |
| --- |
| /\*\*  \* @Title: ${enclosing\_method} <BR>  \* @Description: 请输入你的描述 <BR>  \* @return: ${field\_type} <BR>  \*/ |

注意：使用方式在类上面 输入 **/\*\*** 回车即可

### 新建代码格式样式

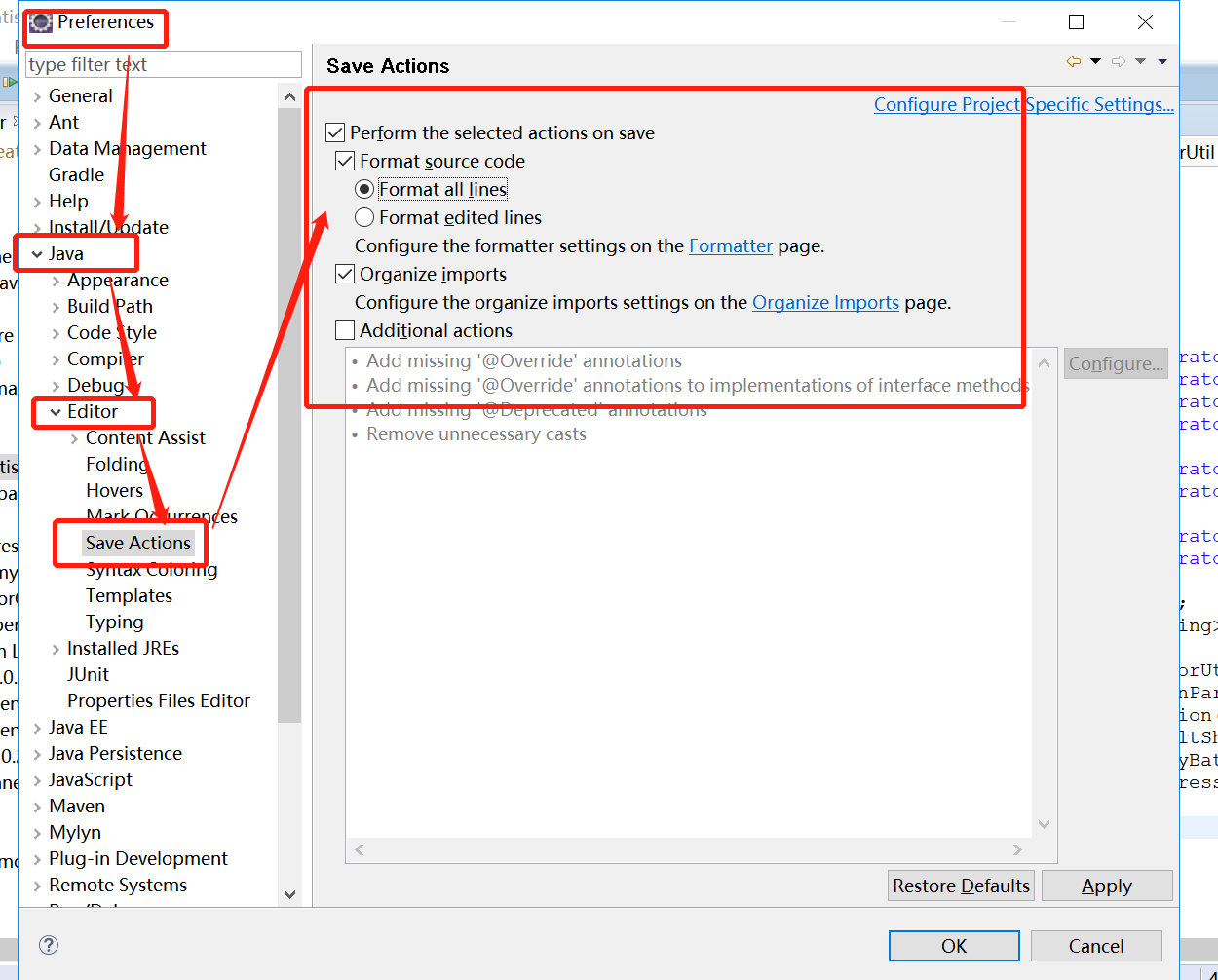


### 导入代码格式化样式





### eclipse设置保存时自动格式化（可选）

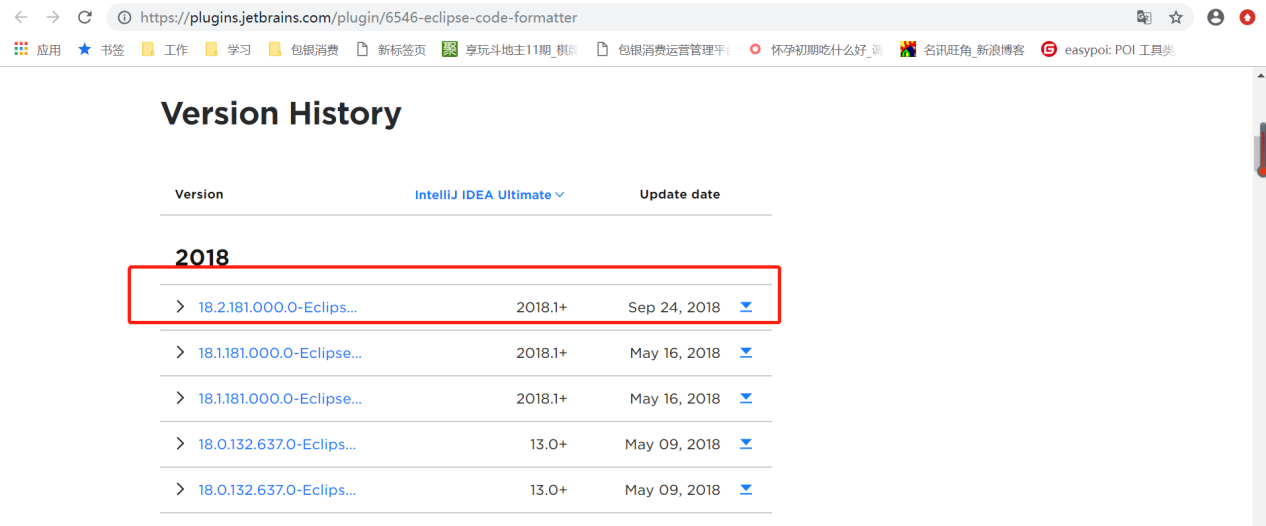


## Idea代码格式规范

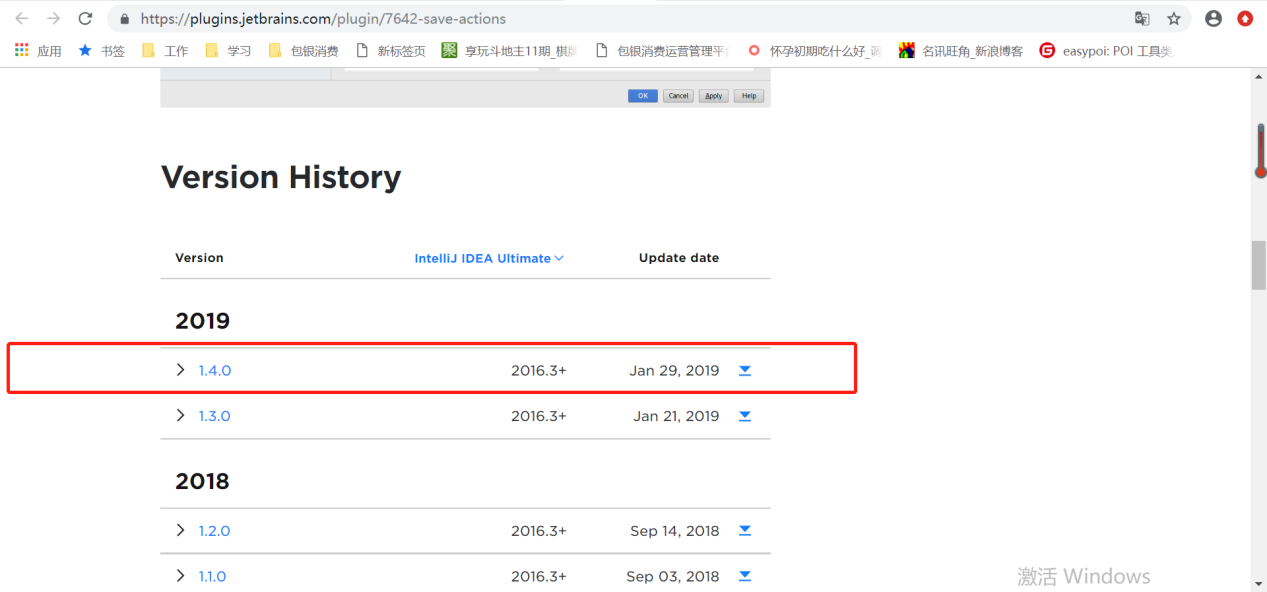
### 下载

【eclipse-code-formatter】：【[https://plugins.jetbrains.com/plugin/6546-eclipse-code-formatter](https://plugins.jetbrains.com/plugin/6546-eclipse-code-formatter" \t "http://www.cnblogs.com/lsysy/p/_blank)】

上为下载的地址，打开后选择一个版本进行下载，本文档使用的为最新的版本：



【save-actions-plugin】：【[https://plugins.jetbrains.com/plugin/7642-save-actions](https://plugins.jetbrains.com/plugin/7642-save-actions" \t "http://www.cnblogs.com/lsysy/p/_blank)】



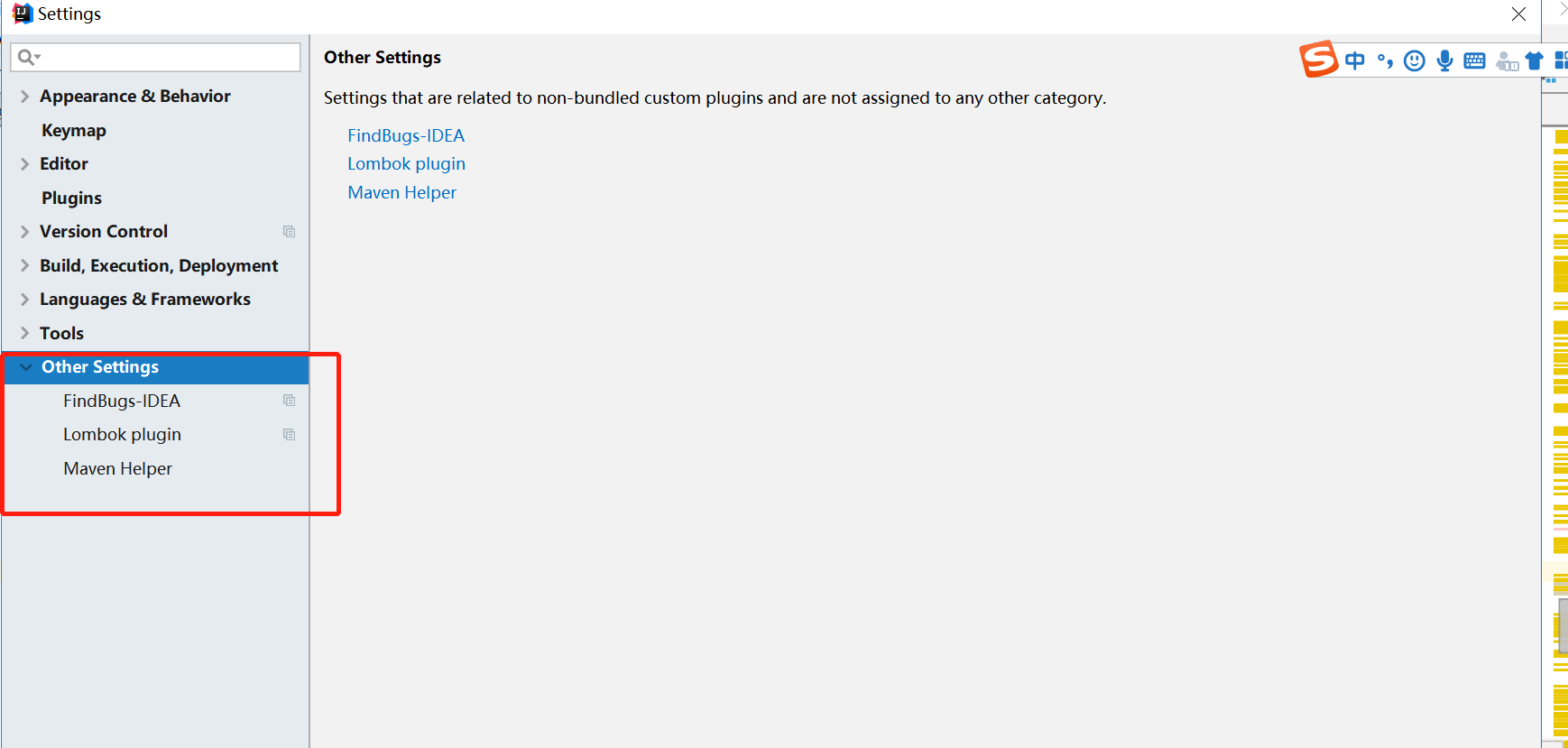
下载后文件如下图:





### eclipseFormatter插件安装

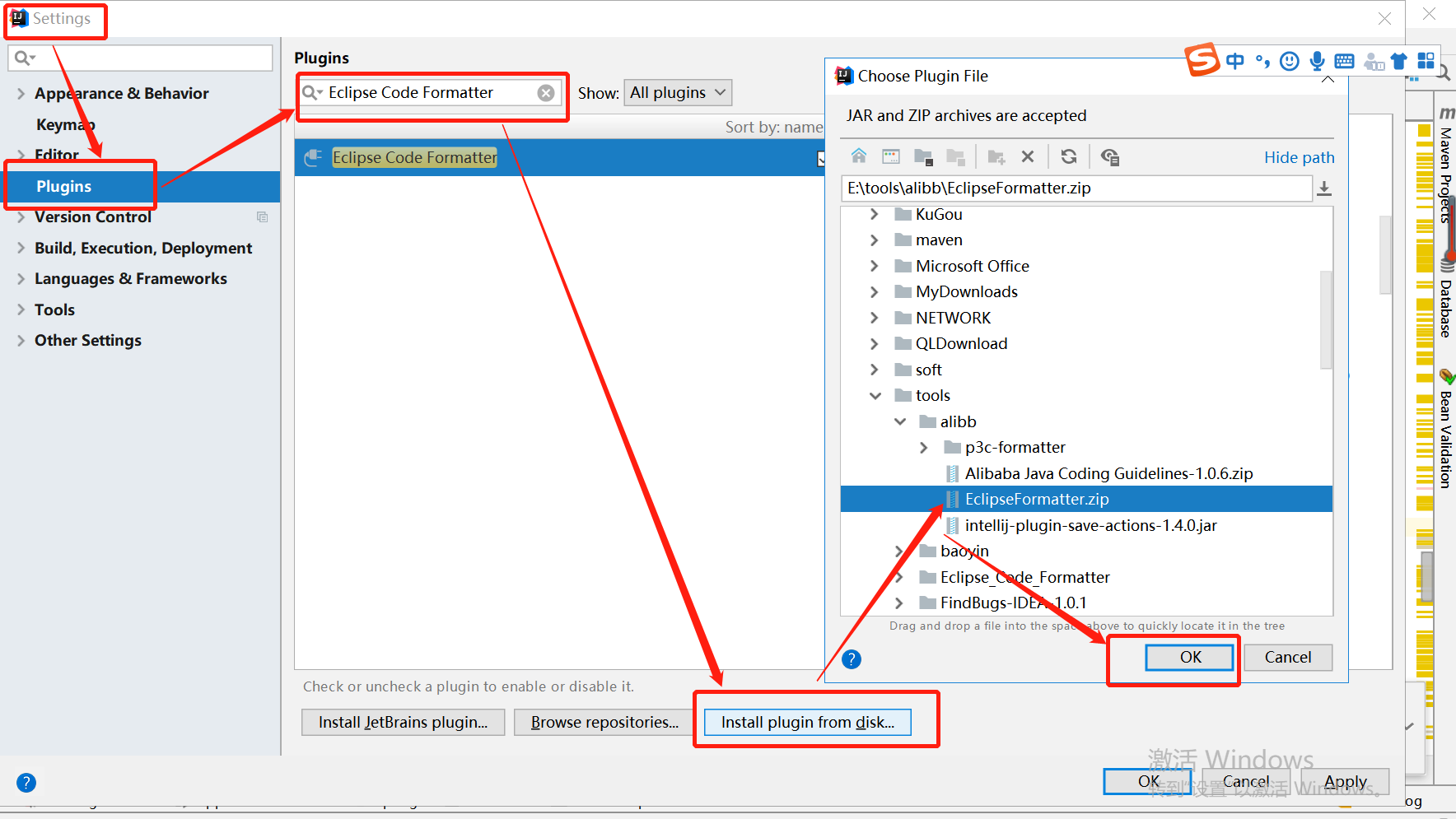
安装插件前,idea只有下图三个插件：

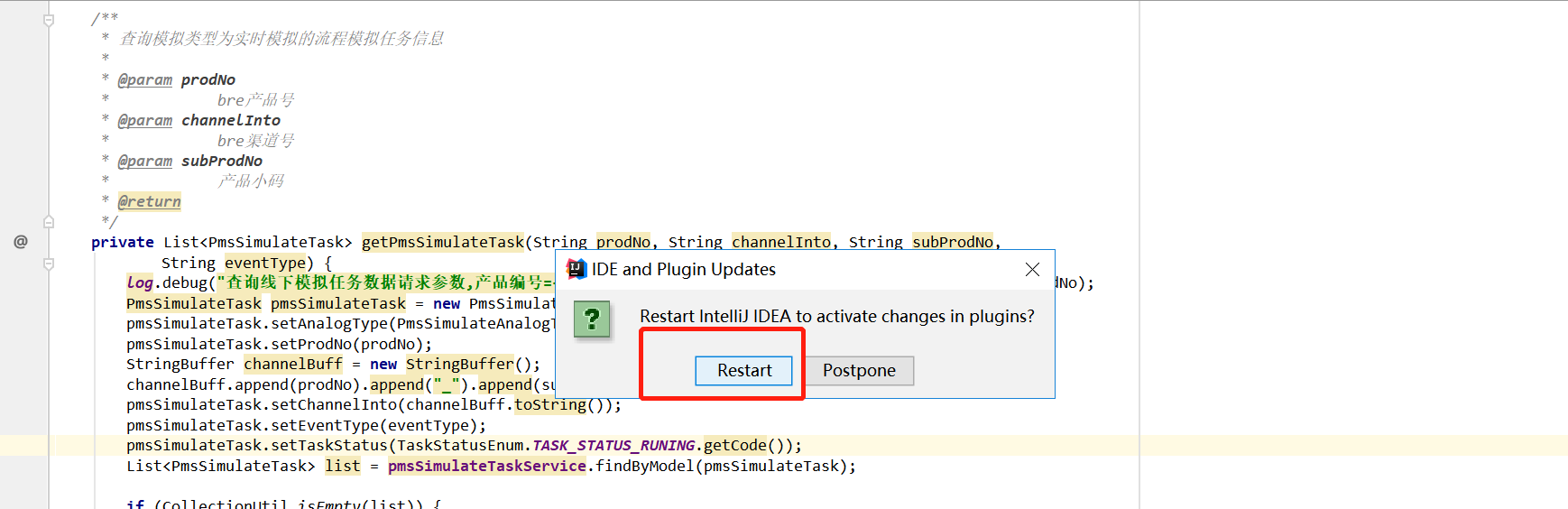


安装步骤:

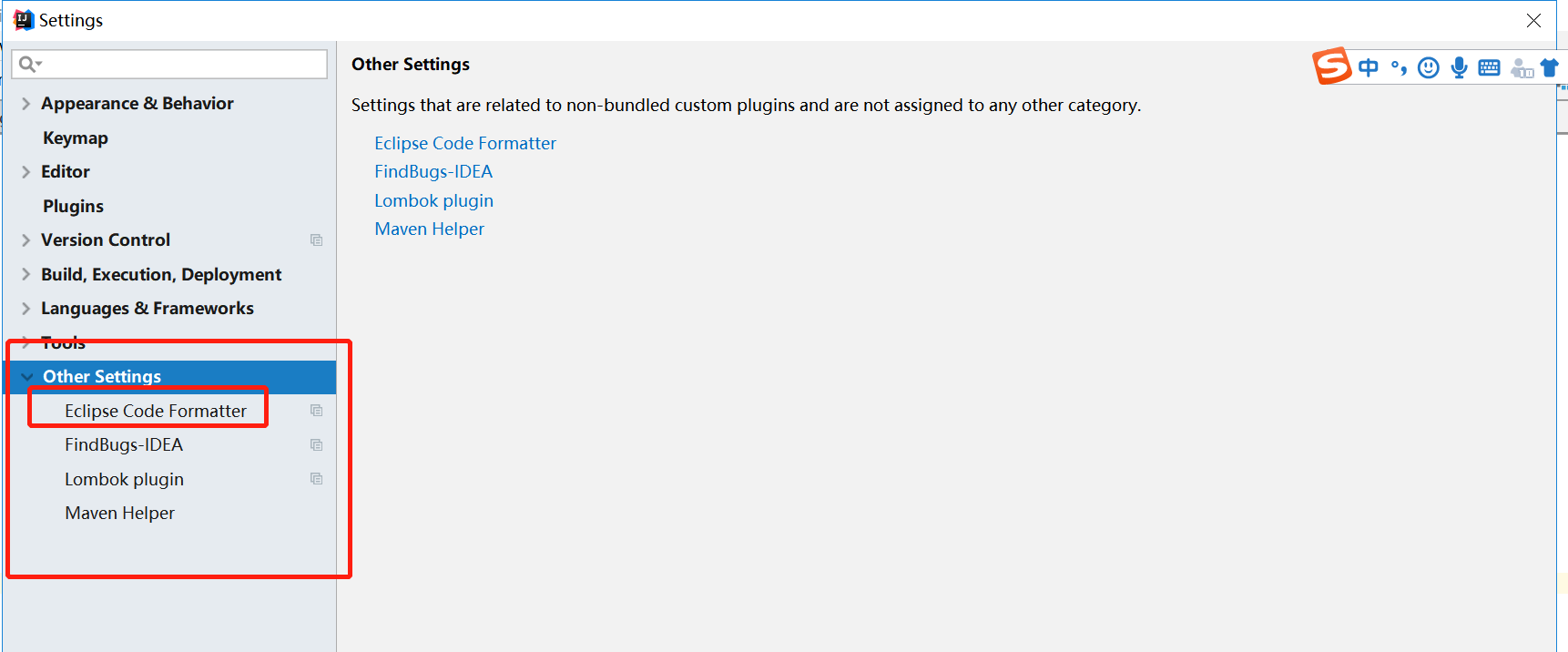
1. 在idea-->File-->Setting-->Plugins 中输入Eclipse Code Formatter并勾选
2. 点击install plugins from disk....，选择4.2.1节中已经下载的EclipseFormatter.zip
3. 点击OK重启生效

安装步骤截图如下:





安装后插件截图如下则表明插件安装成功



### Save Actions插件安装

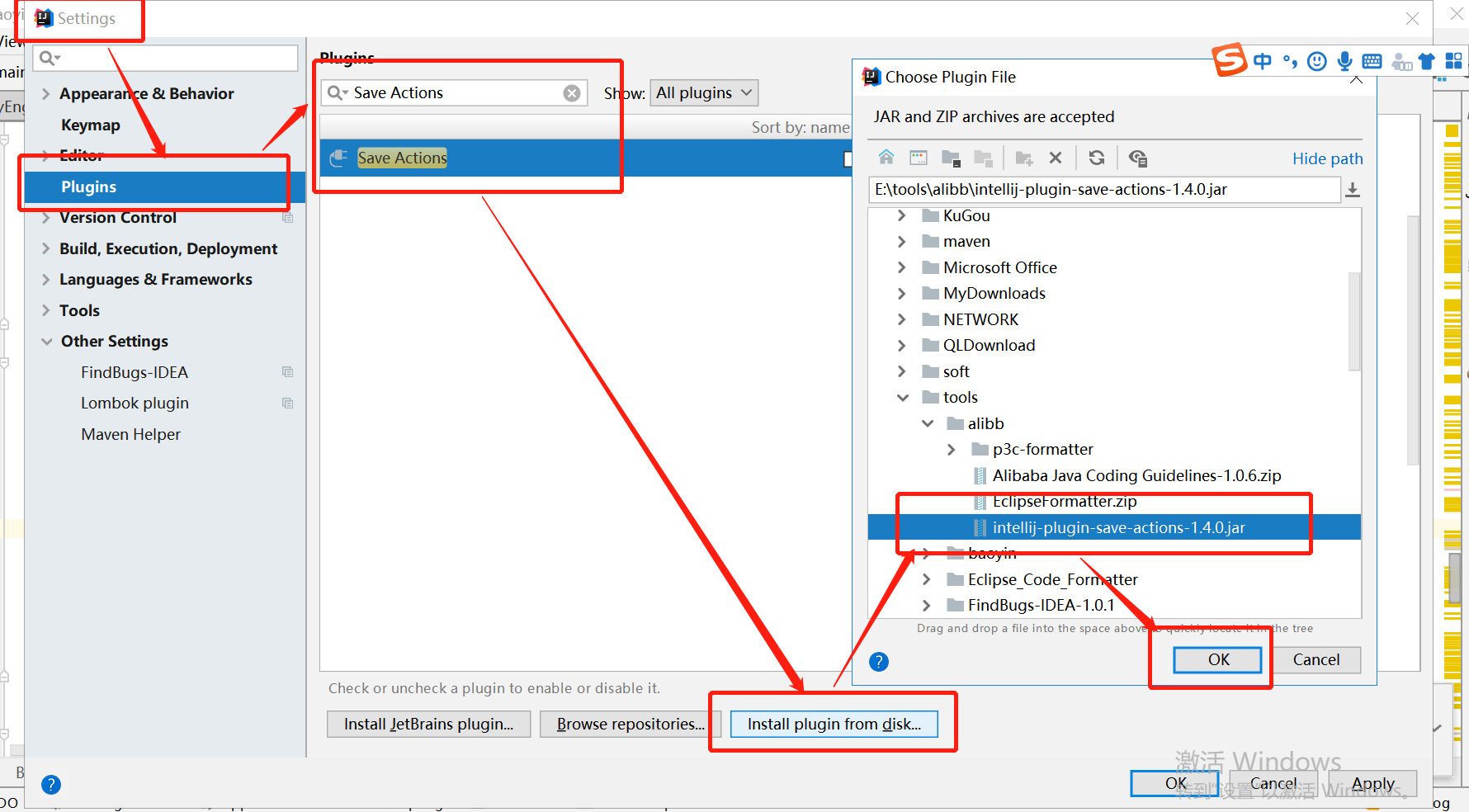
安装步骤:

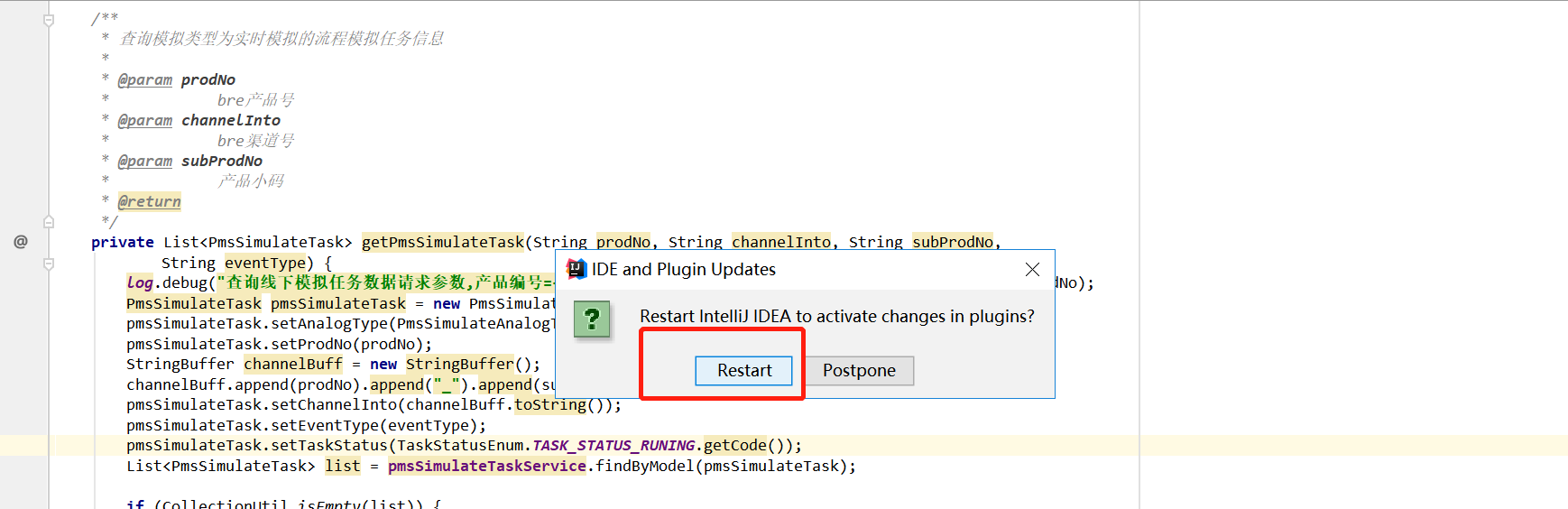
1、在idea-->File-->Setting-->Plugins 中输入Save Actions并勾选

2、点击install plugins from disk，选择4.2.1节中下载的intellij-plugin-save-actions-1.4.0.jar

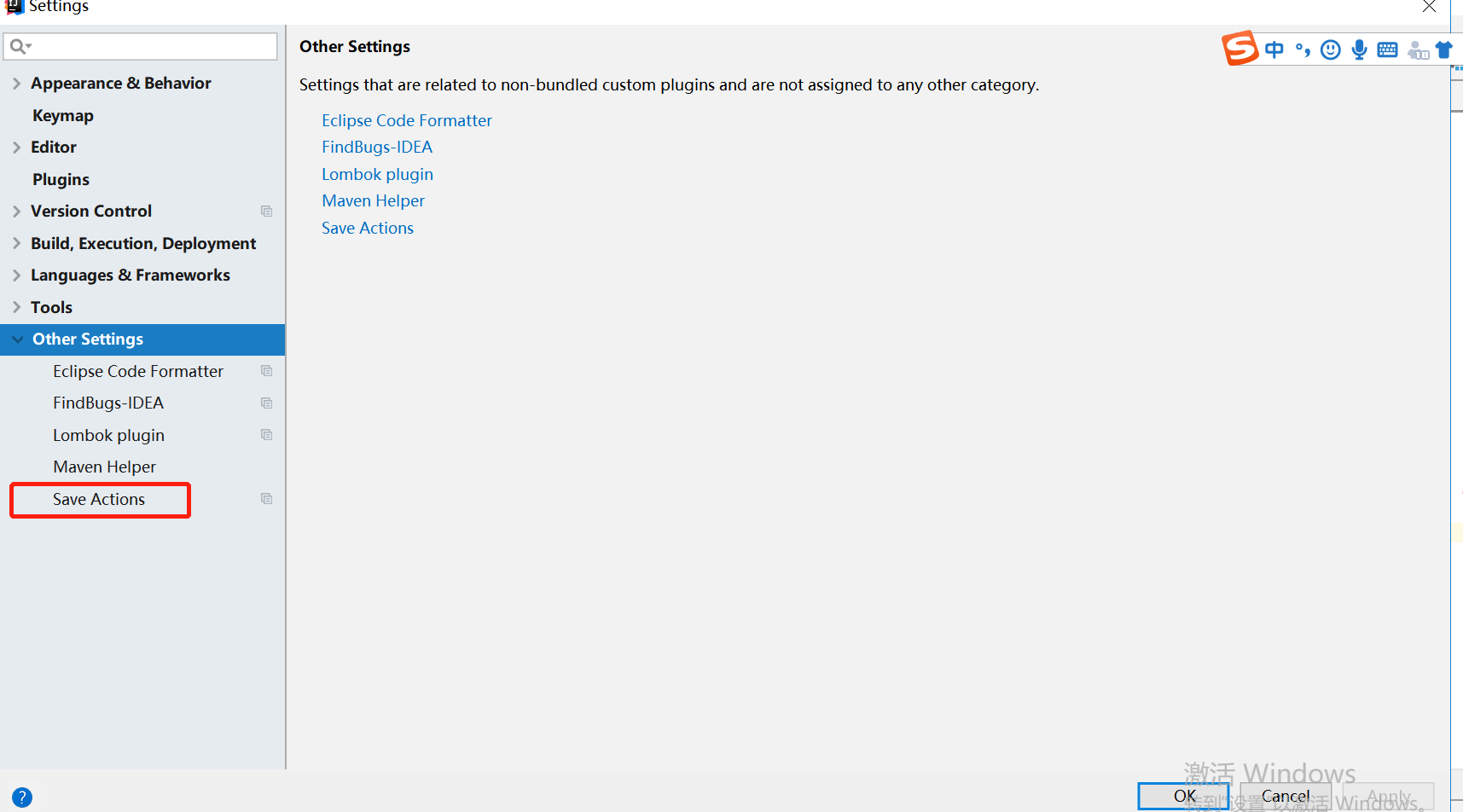
3、点击OK重启生效

安装步骤截图如下





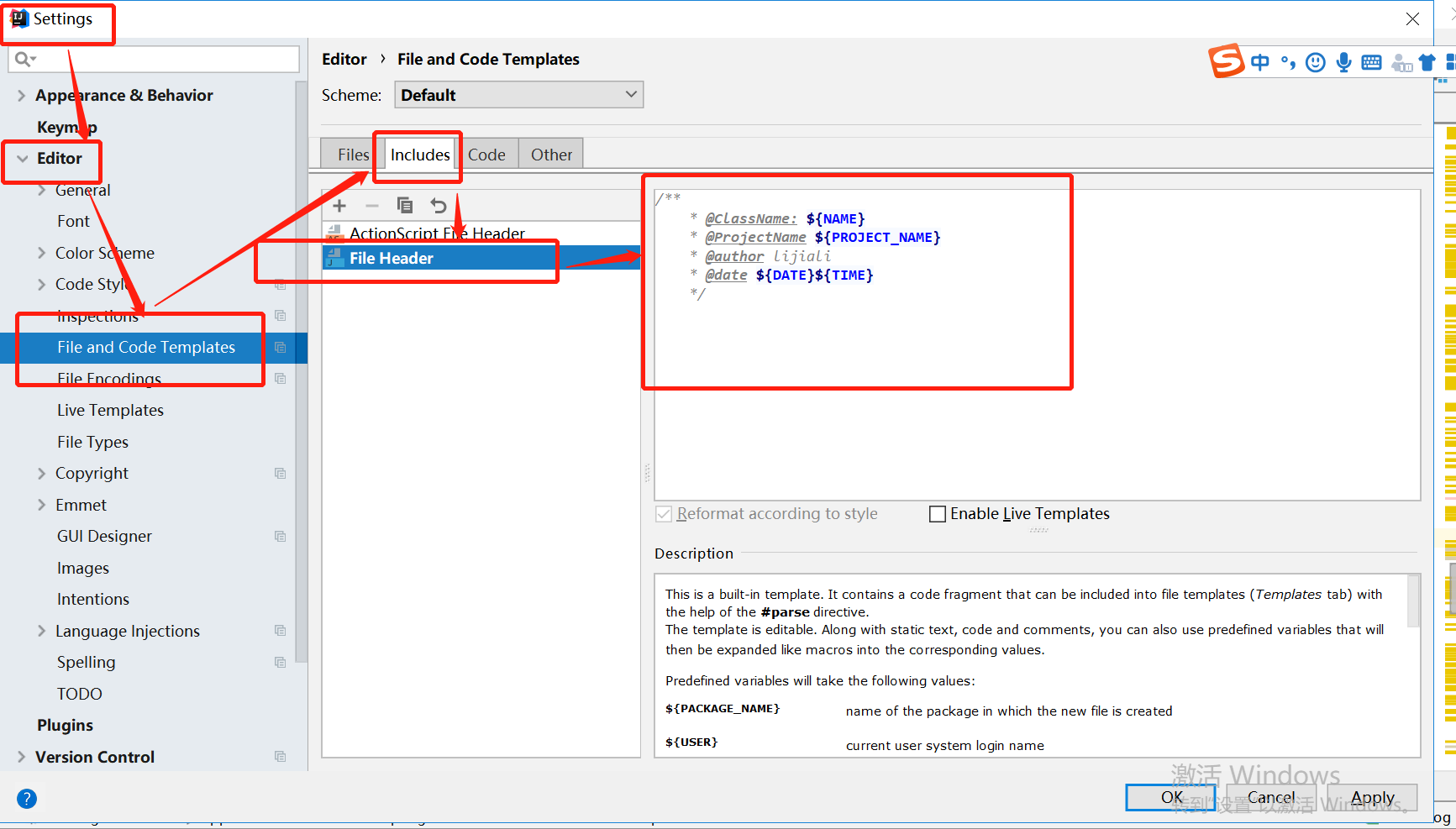
安装后插件截图如下则表明插件安装成功



### 注释模板配置

1)类注释

|  |
| --- |
| /\*\*   \* @ClassName: **${NAME}** \* @ProjectName **${PROJECT\_NAME}** \* @author lijiali  \* @date **${DATE}${TIME}** \*/ |

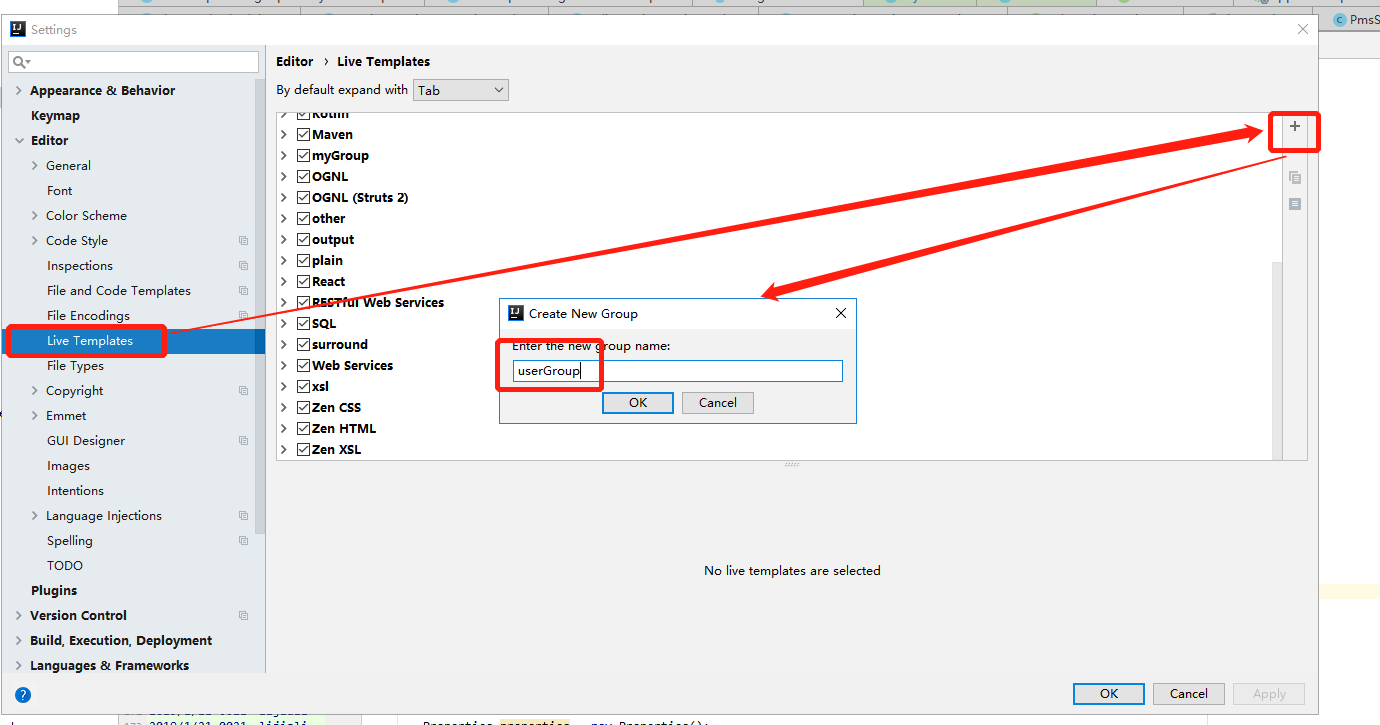


1. 方法注释

IDEA还没有智能到自动为我们创建方法注释，这就是要我们手动为方法添加注释，使用Eclipse时我们生成注释的习惯是

/\*\*+Enter，这里我们也按照这种习惯来设置IDEA的方法注释

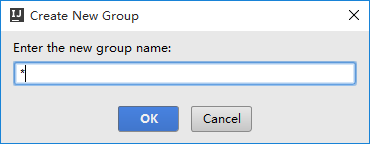
1 File-->Settings-->Editor-->Live Templates



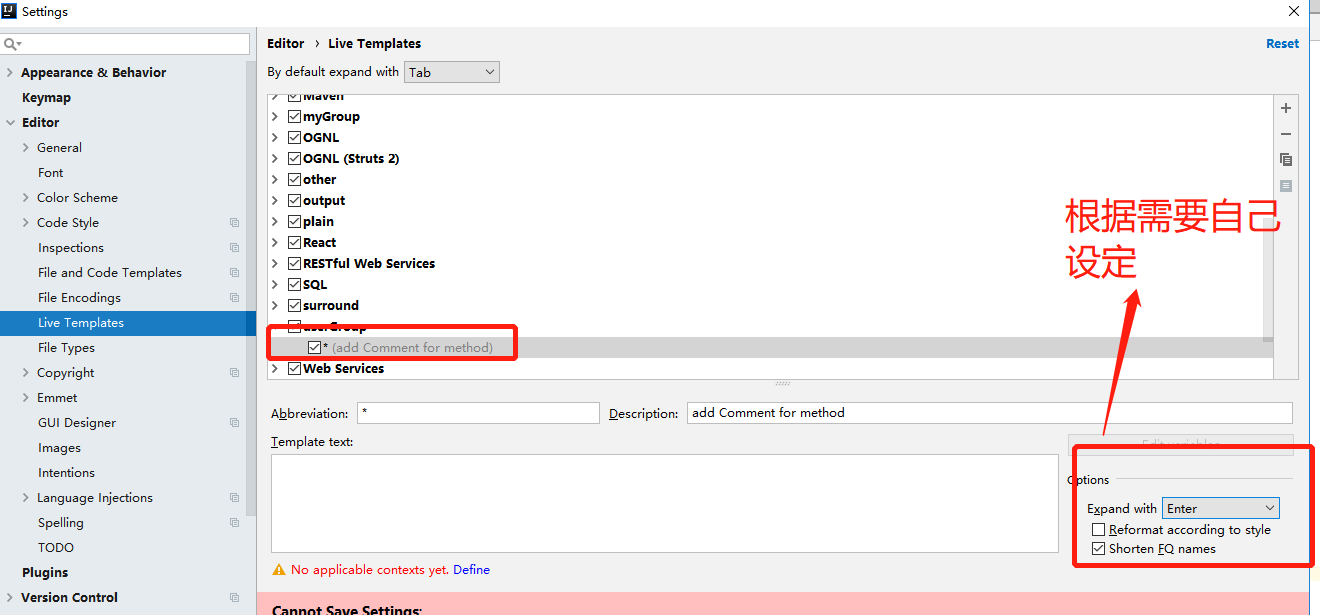
2 新建模板：命名为\*

因为IDEA生成注释的默认方式是：/\*+模板名+快捷键（比如若设置模板名为add快捷键用Tab，则生成方式为/\*add+Tab），如果不采用这样的生成方式IDEA中没有内容的方法将不可用，例如获取方法参数的methodParameters(）

获取方法返回值的methodReturnType(）



3 设置生成注释的快捷键



可以根据需要自己设定，此文档设置为 Enter

3 设置模板：模板内容如下

注意第一行，只有一个\*而不是/\*

在设置参数名时必须用${参数名}$的方式，否则第五步中读取不到你设置的参数名

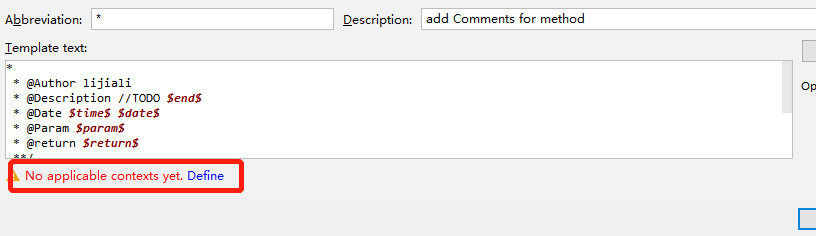
|  |
| --- |
| \*  \* @Author lijiali  \* @Description //TODO $end$  \* @Date $time$ $date$  \* @Param $param$  \* @return $return$  \*\*/ |

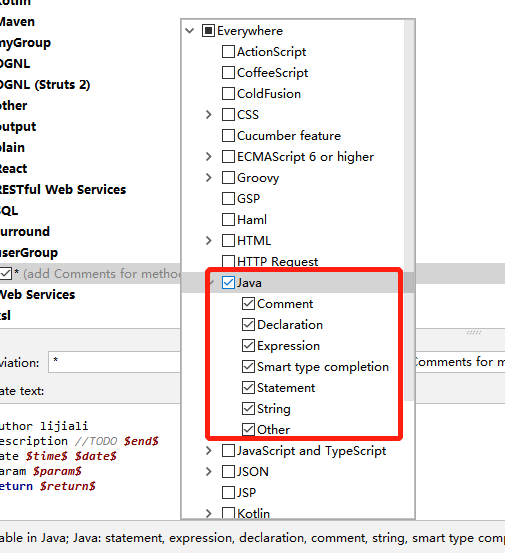
如果使用/\*生成的模板注释将会是如下效果：所以我们要去掉最前面的/\*

|  |
| --- |
| /\*/\*\*  \* @Author lijiali  \* @Description //TODO $end$  \* @Date $time$ $date$  \* @Param $param$  \* @return $return$  \*\*/ |

5 设置模板的应用场景

点击模板页面最下方的警告，来设置将模板应用于那些场景，一般选择EveryWhere-->Java即可（如果曾经修改过，则显示为change而不是define）

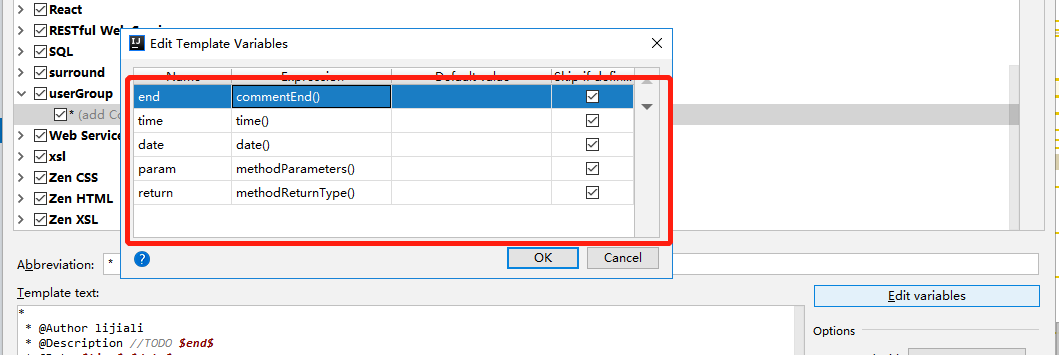




（6）设置参数的获取方式

选择右侧的Edit variables按钮

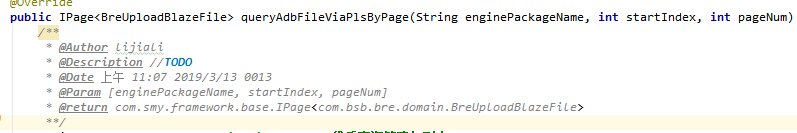




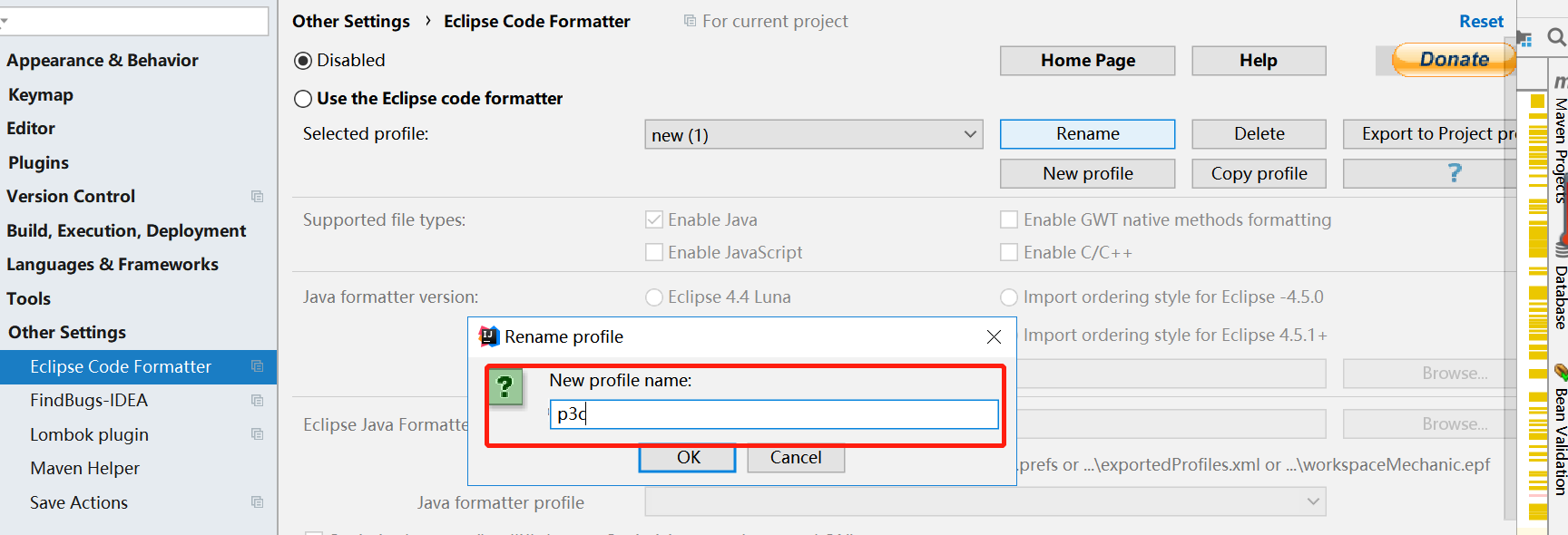
PS:第五步和第六步顺序不可颠倒，否则第六步将获取不到方法

（7）效果图

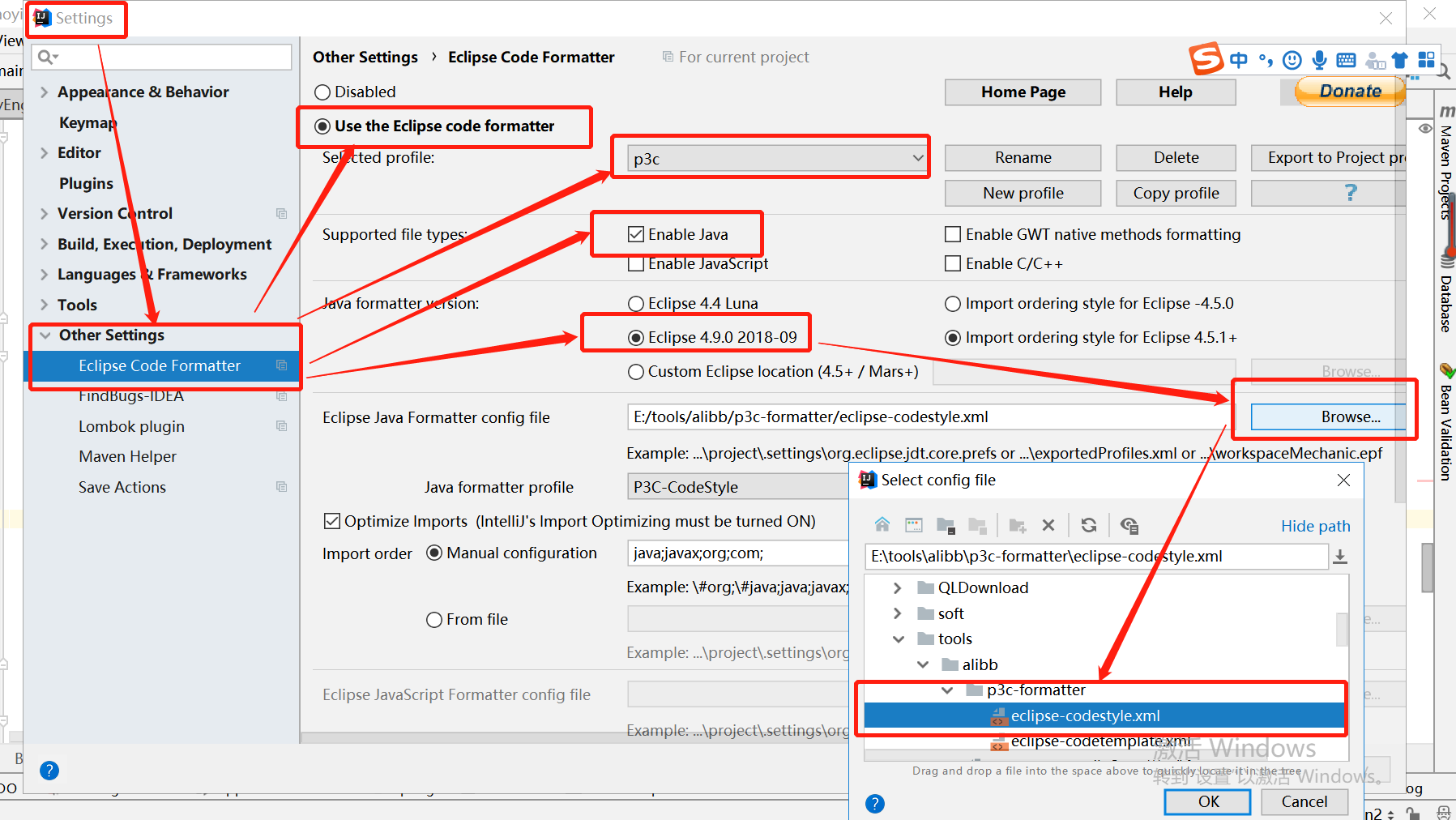
创建方法，在方法上面写：/\*+模板名+Enter-->/\*\*+Enter



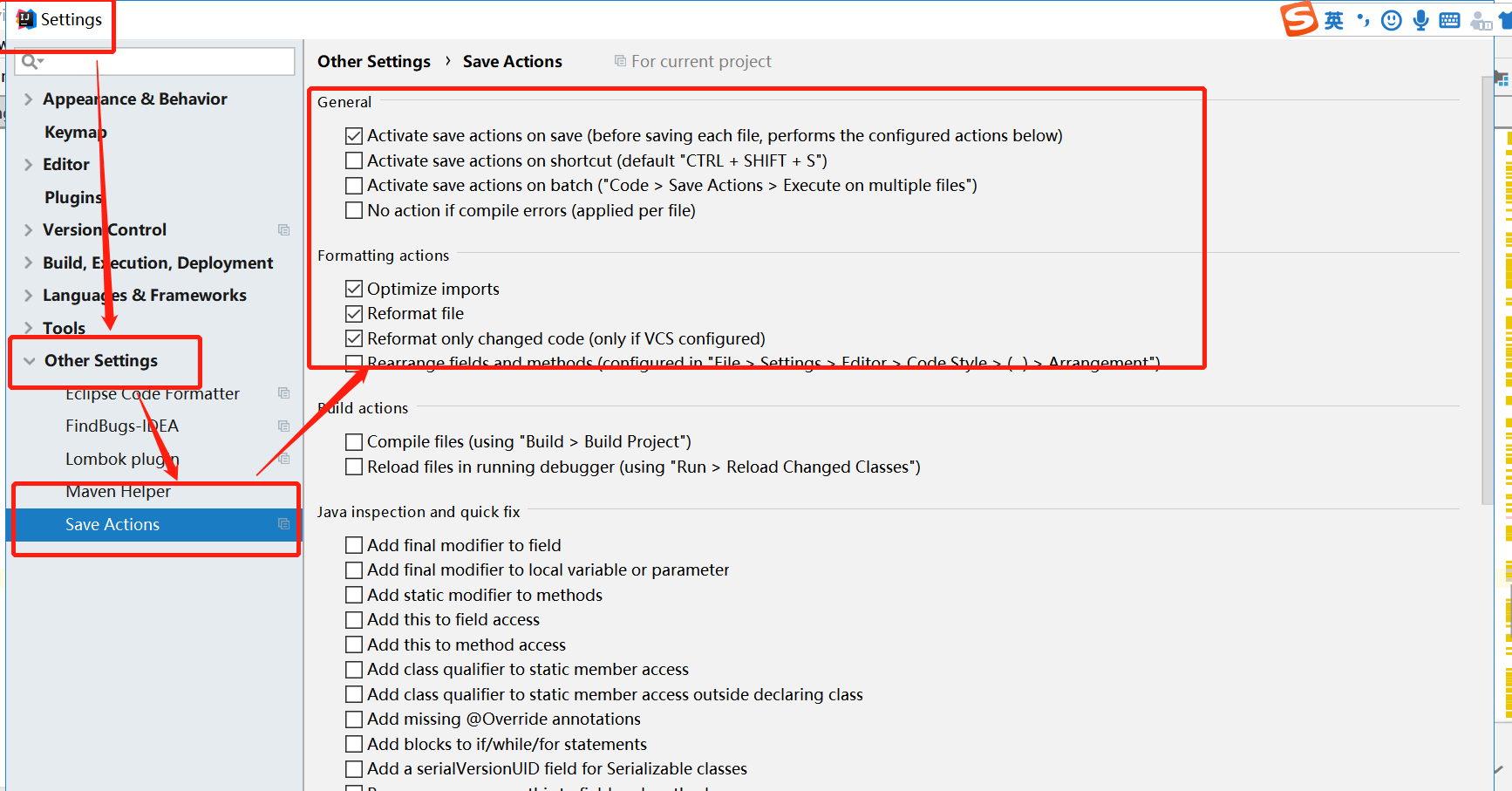
### 新建代码格式样式



### 导入代码格式化样式（导入eclipse-codestyle.xml）



### Idea设置保存时自动格式化（可选）

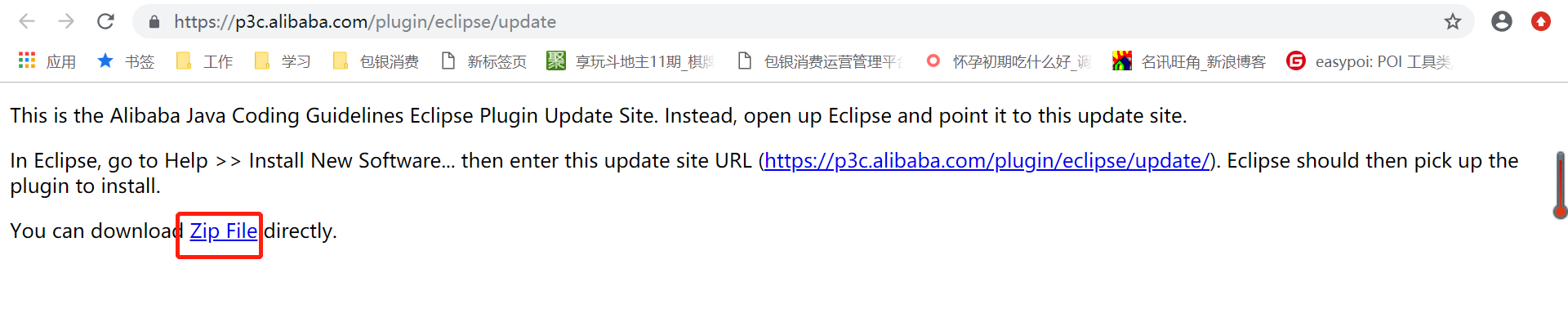


# 阿里编码规约设置

## eclipse编码规约

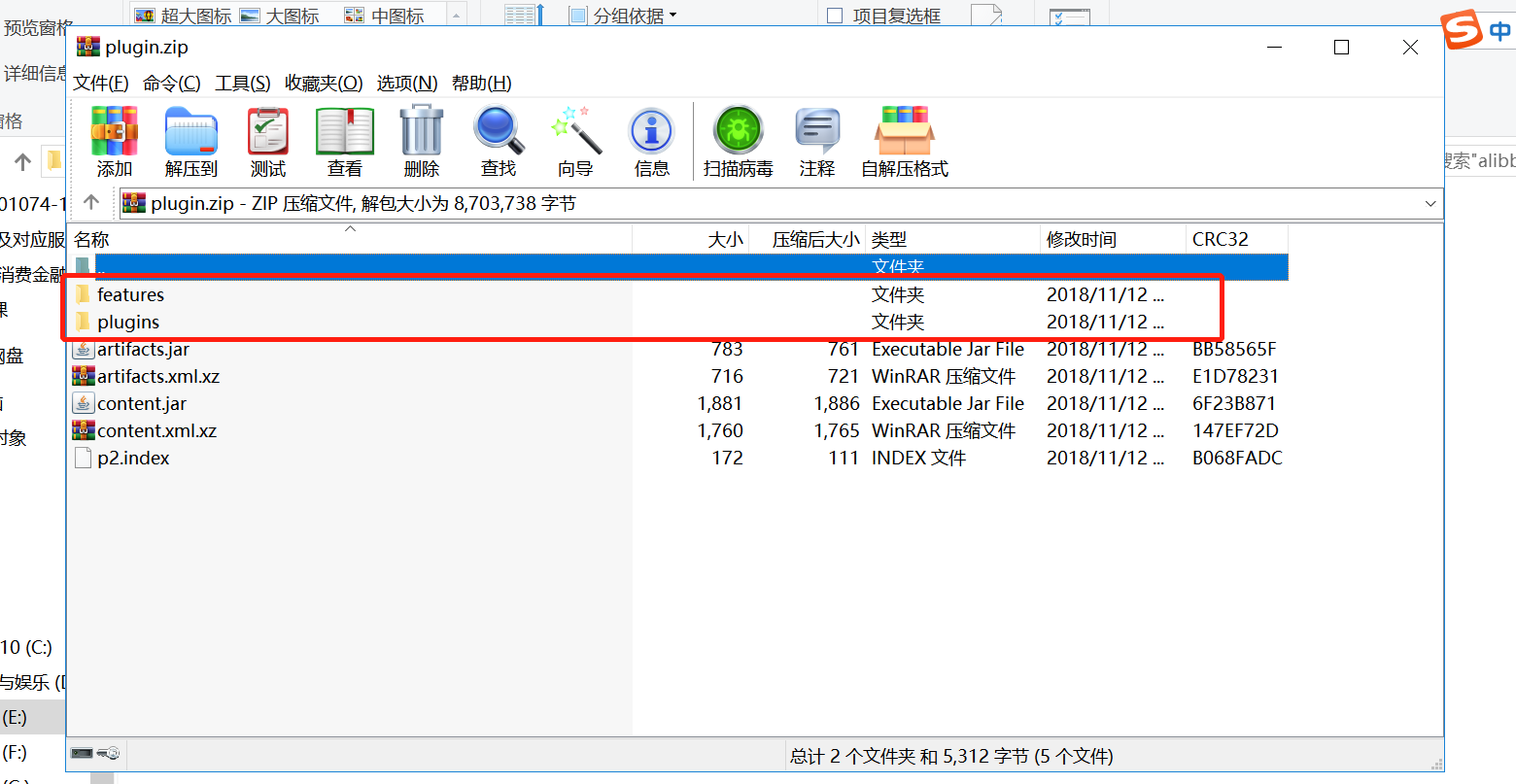
### 插件下载

【p3c-eclipse-plugin】：【[https://p3c.alibaba.com/plugin/eclipse/update](https://p3c.alibaba.com/plugin/eclipse/update" \t "http://www.cnblogs.com/lsysy/p/_blank)】



点击 Zip File进行下载，下载后如下图





### 插件安装

将5.1.2节中解压的【features】及【plugins】复制到eclipse目录，重启eclipse

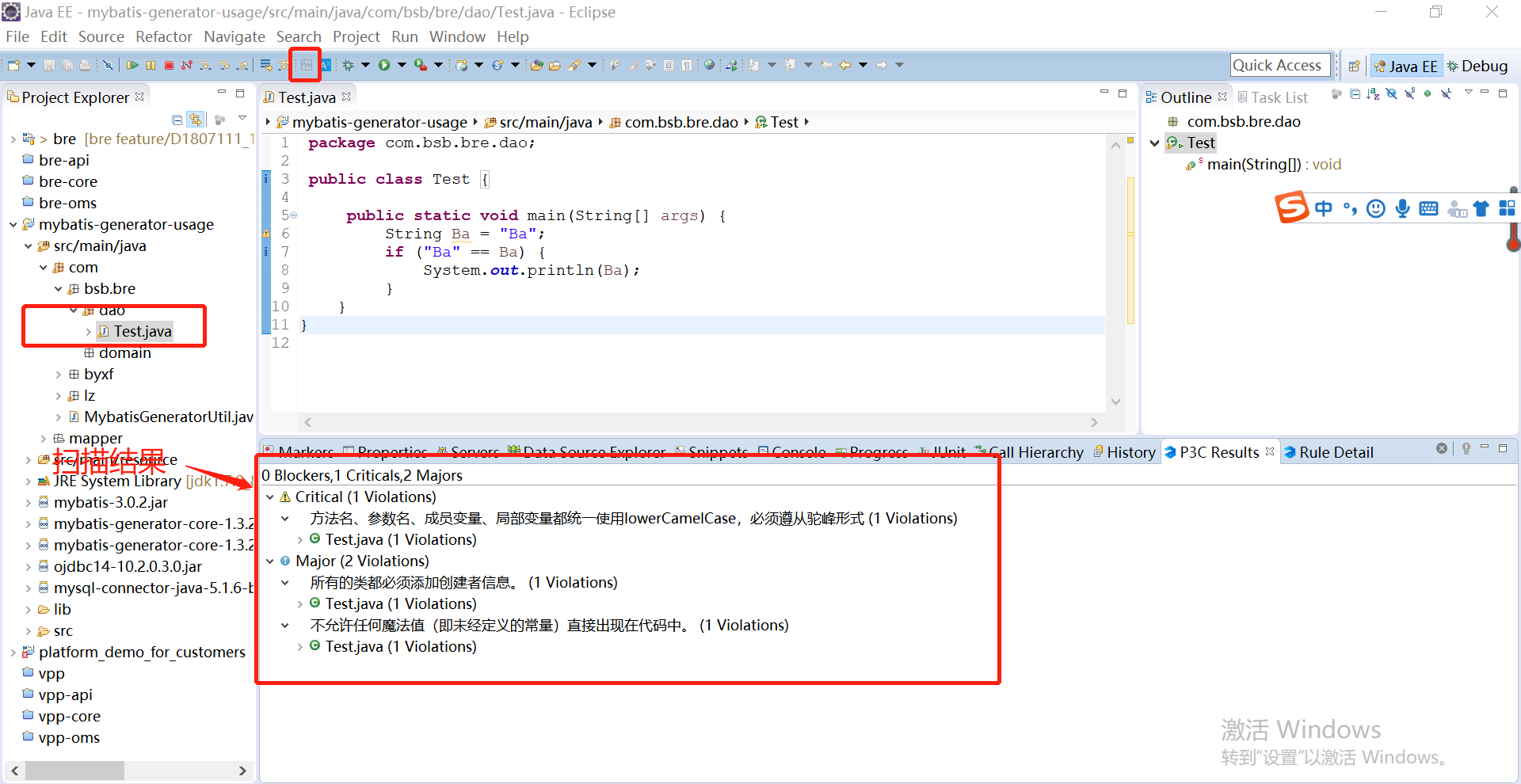
### 编写测试文件

|  |
| --- |
| **public** **class** Test {  **public** **static** **void** main(String[] args) {  String Ba = "Ba";  **if** ("Ba" == Ba) {  System.***out***.println(Ba);  }  }  } |

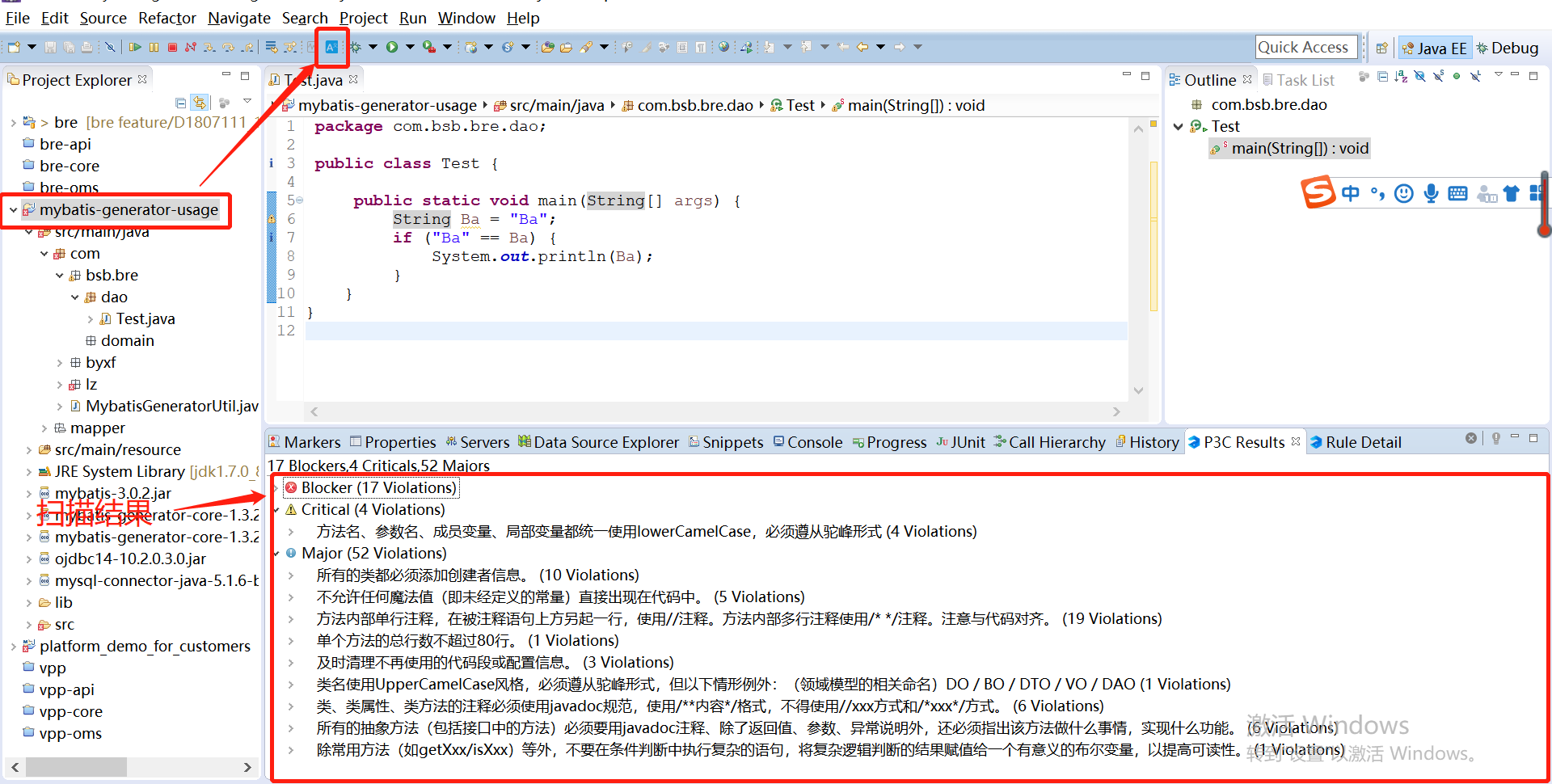
### **点击进行阿里编码规约扫描**

点击IMG_256可进行语言切换，切换完重启即可

1）单个文件扫描,选择该文件然后点击IMG_256或者右键选择扫描功能进行扫描



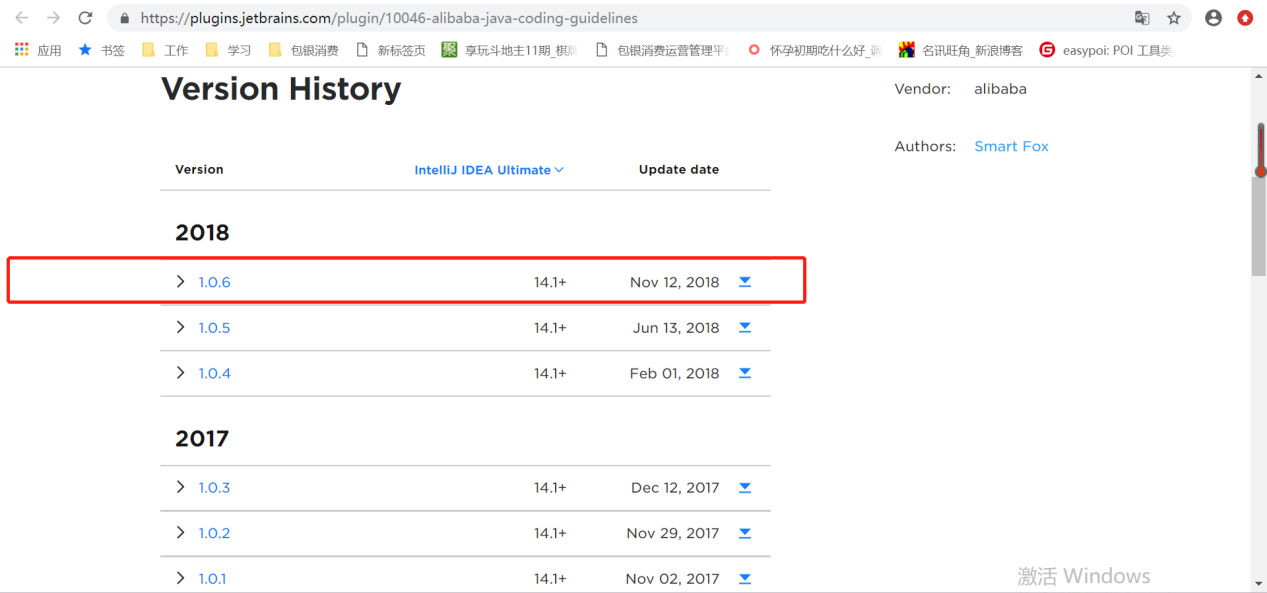
2）多个文件扫描,先选择项目目录然后点击IMG_256进行扫描即可



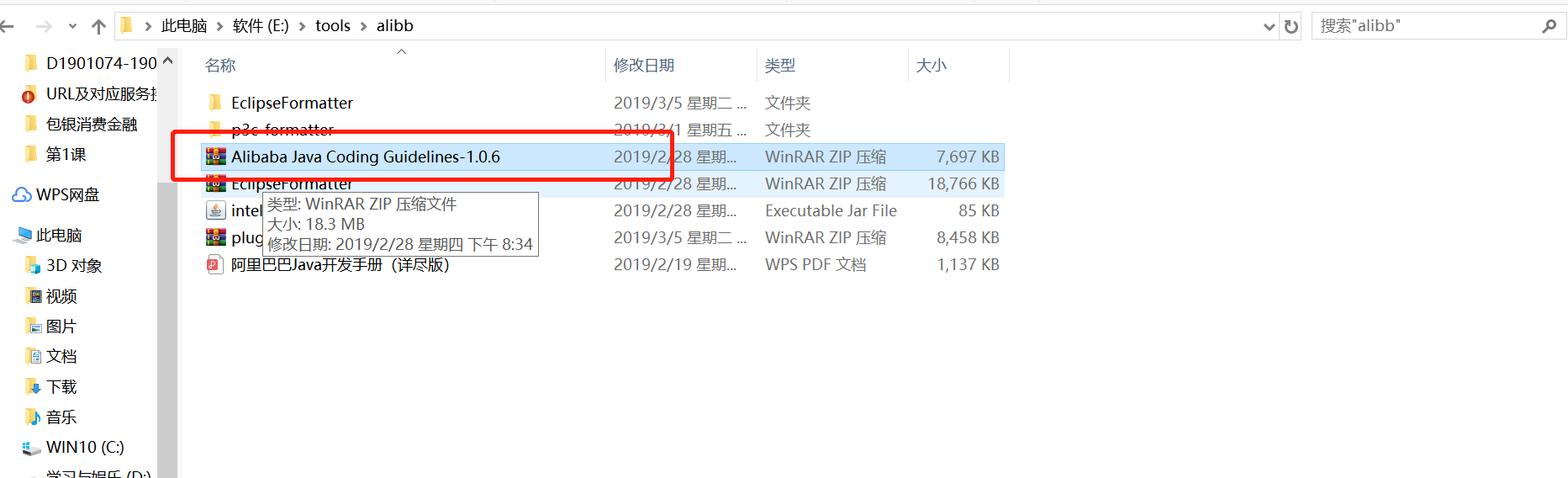
## Idea编码规约

### 插件下载

【p3c-idea-plugin】：【[https://plugins.jetbrains.com/plugin/10046-alibaba-java-coding-guidelines](https://plugins.jetbrains.com/plugin/10046-alibaba-java-coding-guidelines" \t "http://www.cnblogs.com/lsysy/p/_blank)】



点击下载按钮进行下载，下载后如下图



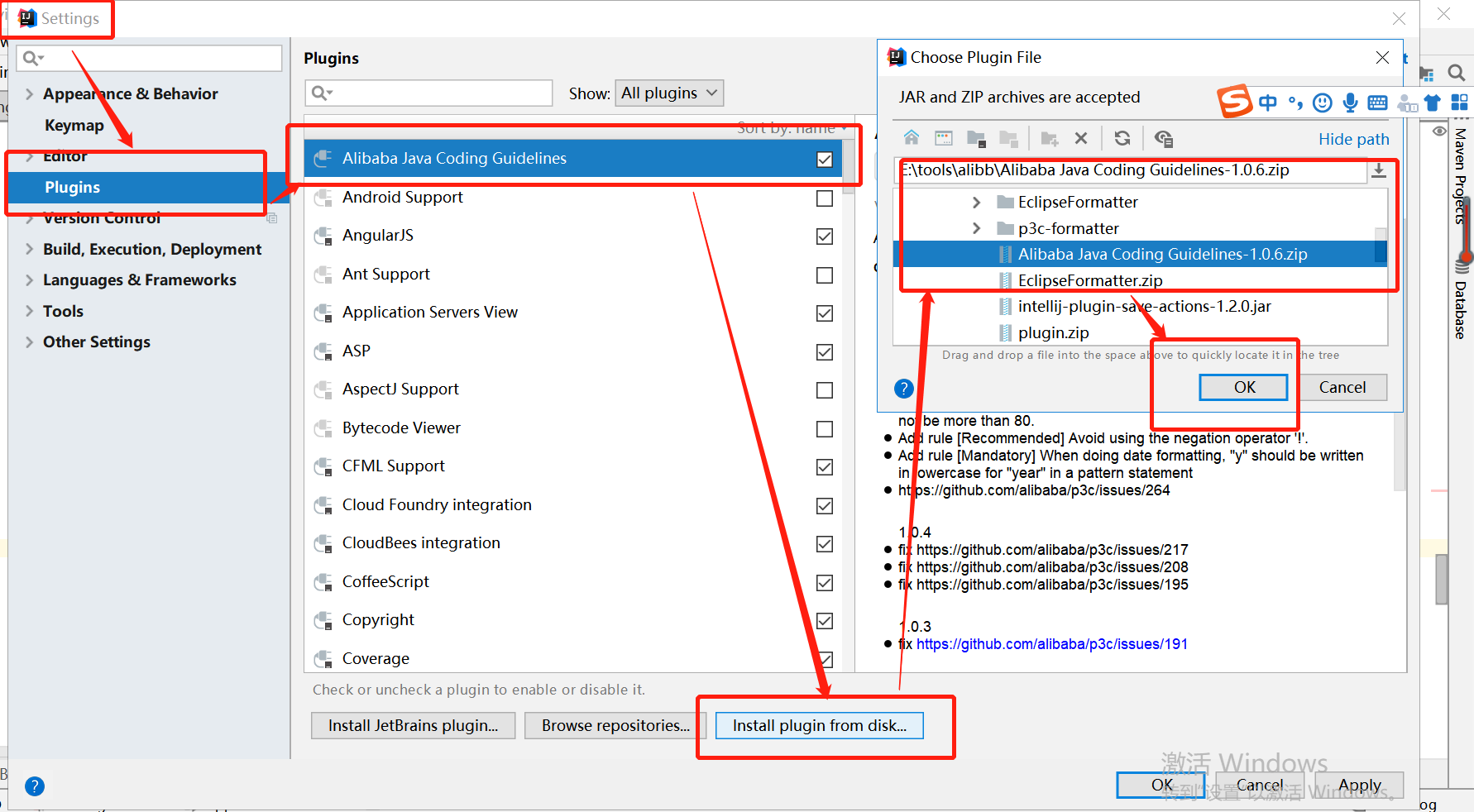


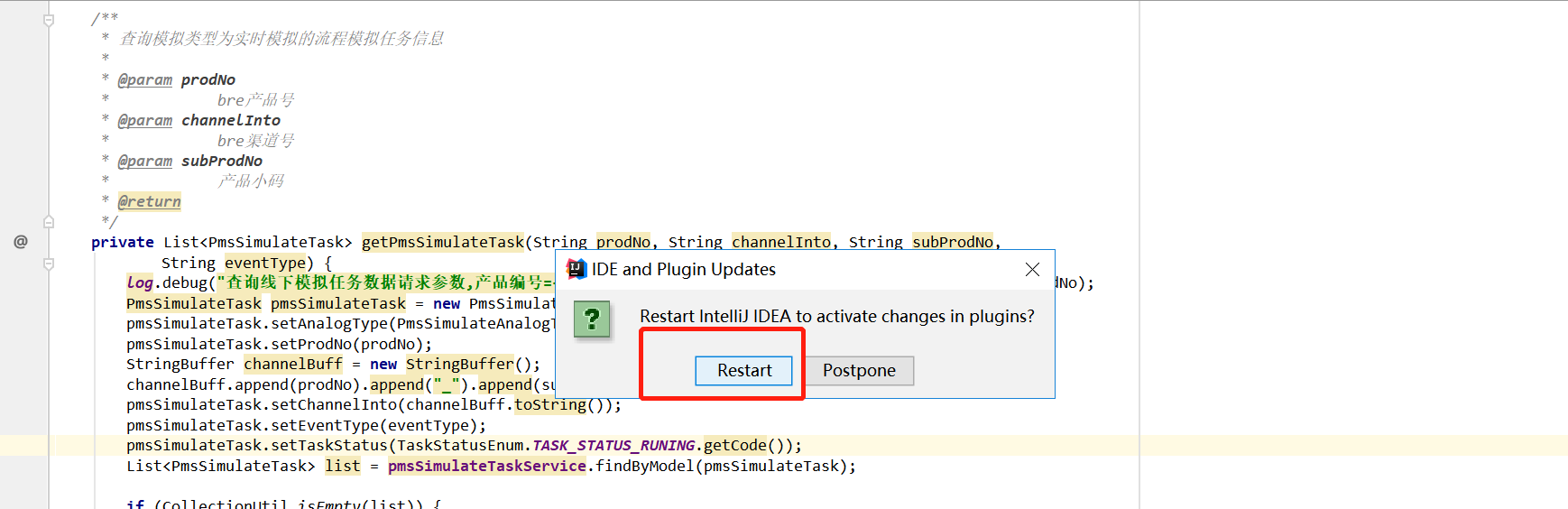
### 插件安装

安装步骤:

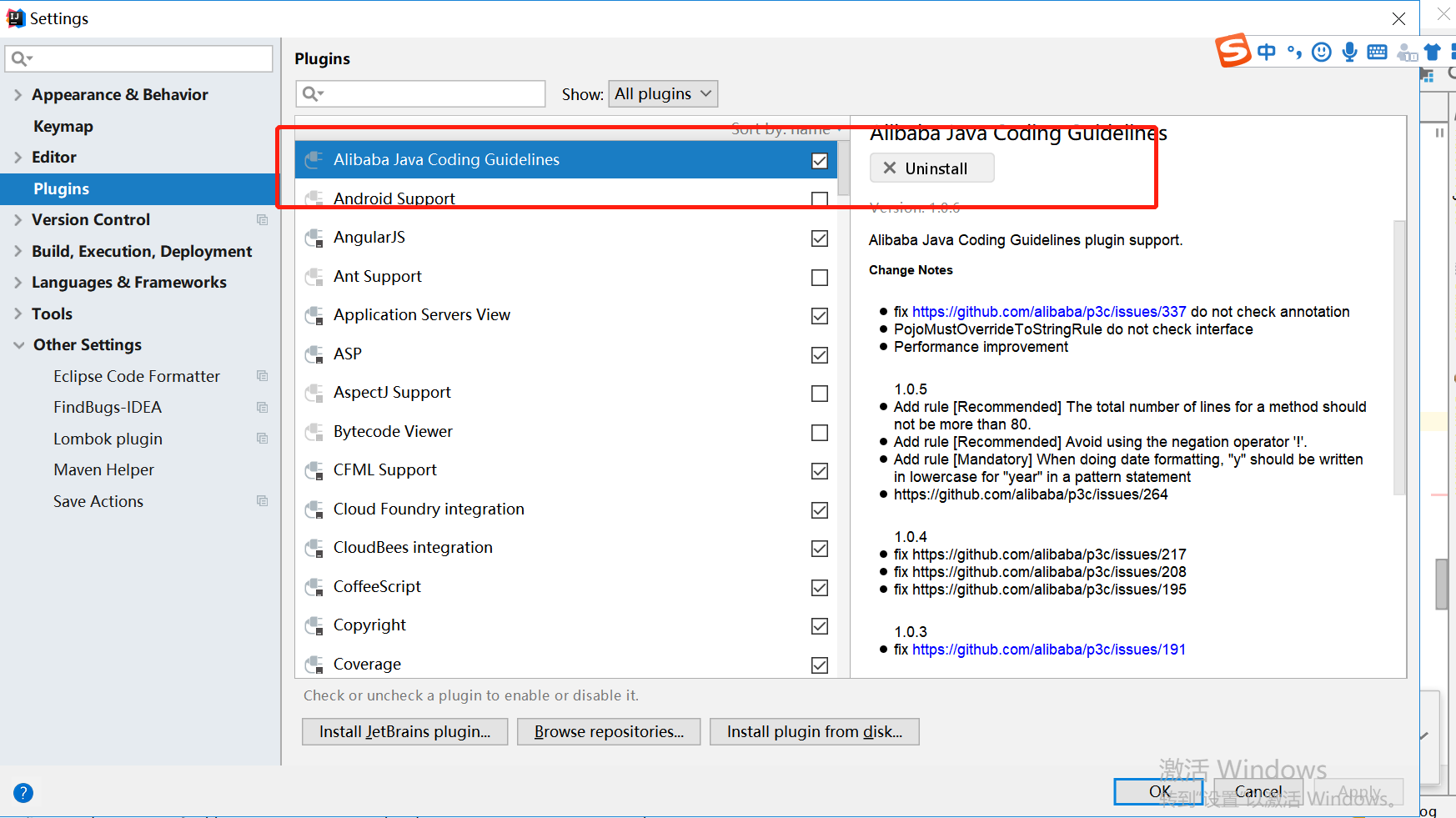
1. 在idea-->File-->Setting-->Plugins 中输入Eclipse Code Formatter并勾选
2. 点击install plugins from disk....，选择4.2.1节中已经下载的EclipseFormatter.zip
3. 点击OK重启生效

安装步骤截图如下:



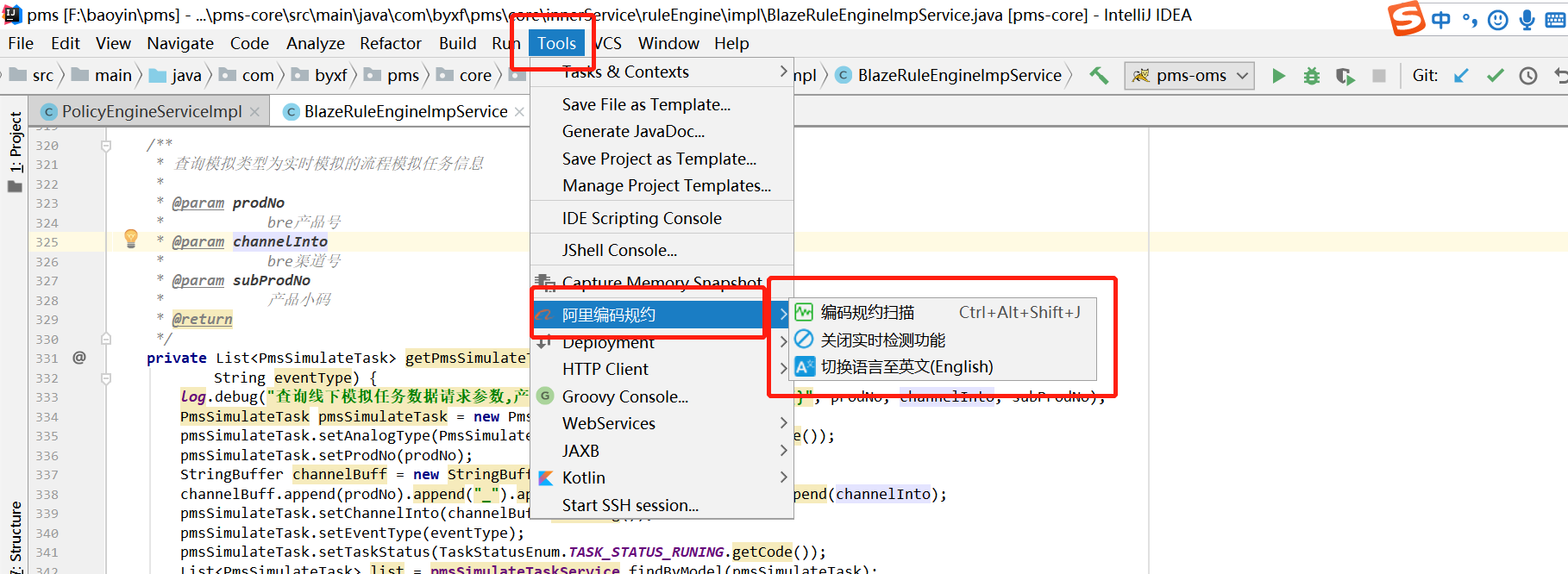


安装后插件截图如下则表明插件安装成功



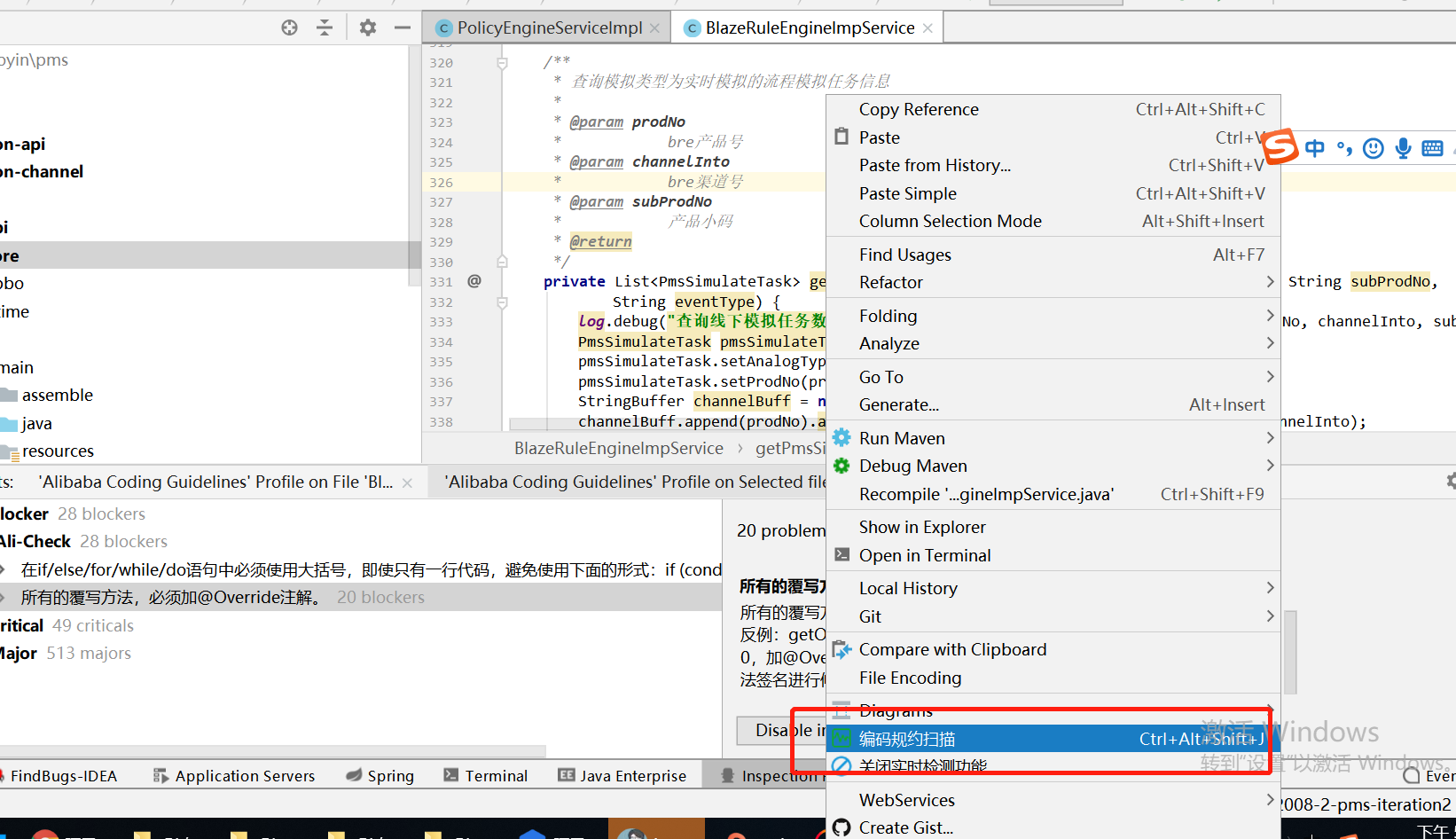
### 插件使用

点击菜单栏中Tools-->阿里编码规约-->切换语言/关闭或打开实时检测功能/编码规约扫描



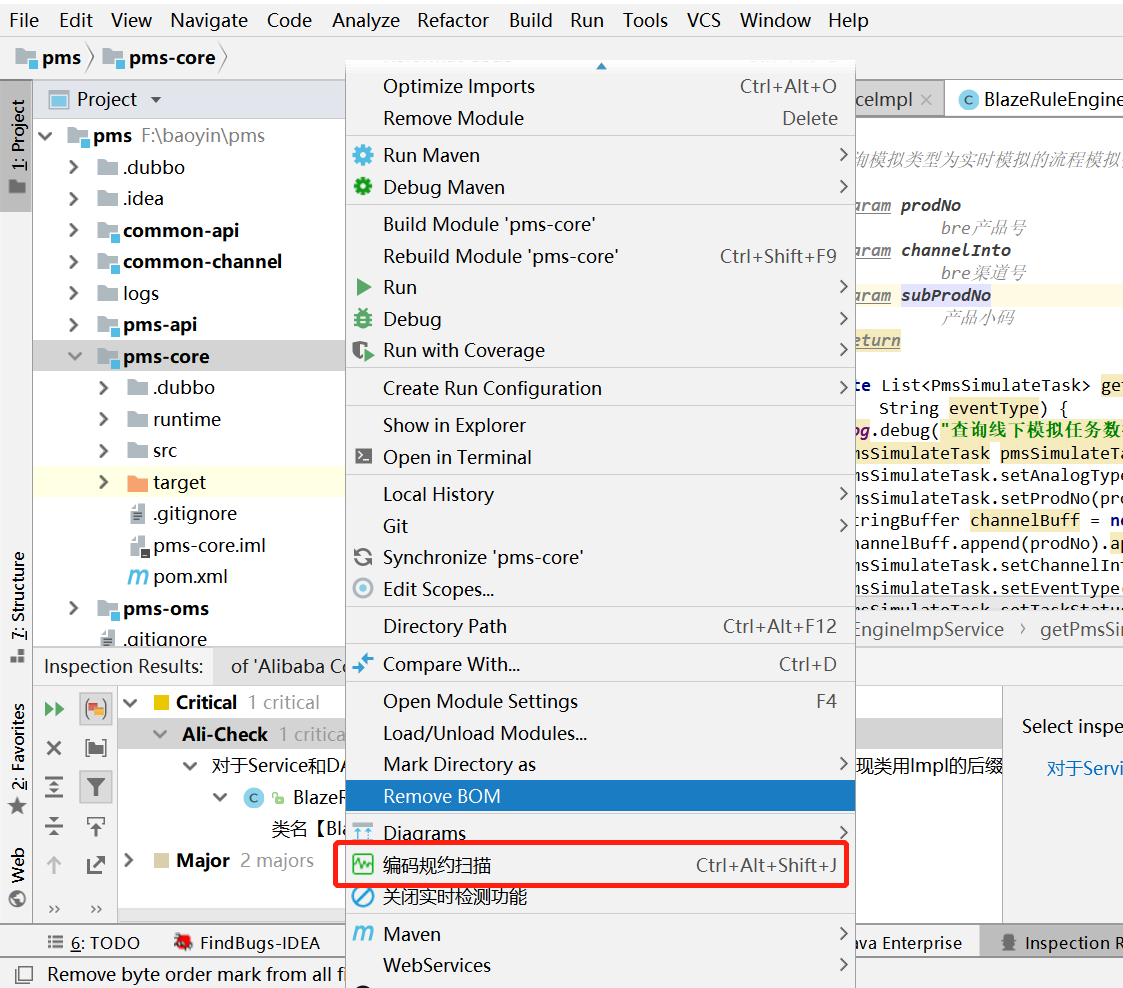
1）单文件扫描

在需要扫描的文件中单击右键点击扫描案件即可



2）多文件扫描

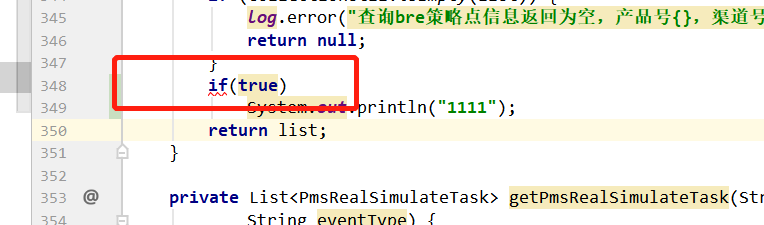
在项目目录或者在工程目录右键会出现检测的功能按钮,点击后会对整个目录或工程进行扫描



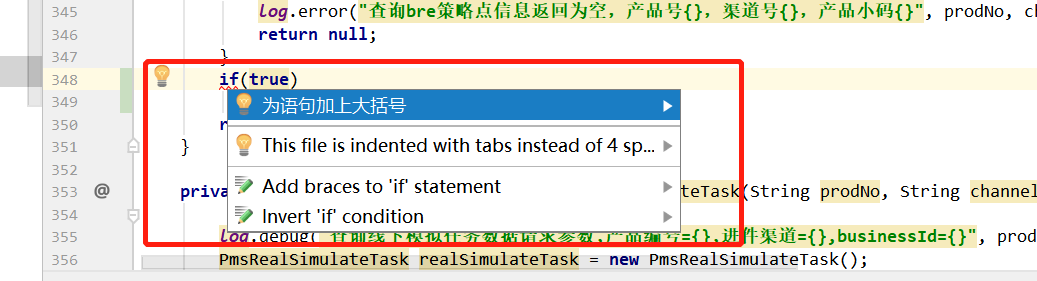


3）实时扫描

前面说的两种方式是手动检测，插件也提供了实时监测的功能，此功能默认开启，一旦开启则会在你编写代码时就会实时监测，一旦有不符合规范的语句就会出现错误提示。



由于大括号不规范的问题，if下方有错误红线的提示。



idea中有quick fix快捷键:option/alt+enter,点击后则可以自动添加大括号。

# 代码扫描工具-findBugs

## 简介

Findbugs是一个静态分析工具，它检查类或者jar文件，将字节码与一组缺陷模式进行对比以发现可能的问题。利用这个工具可以在不实际运行程序的情况下对软件进行分析。可以帮助改进代码质量。Findbugs提供了方便操作的可视化界面，同时也可以作为eclipse的一个插件来使用，而我们是用的最多的还是作为eclipse插件来使用。

Findbugs的常用操作,Findbugs找出的bug有三种颜色

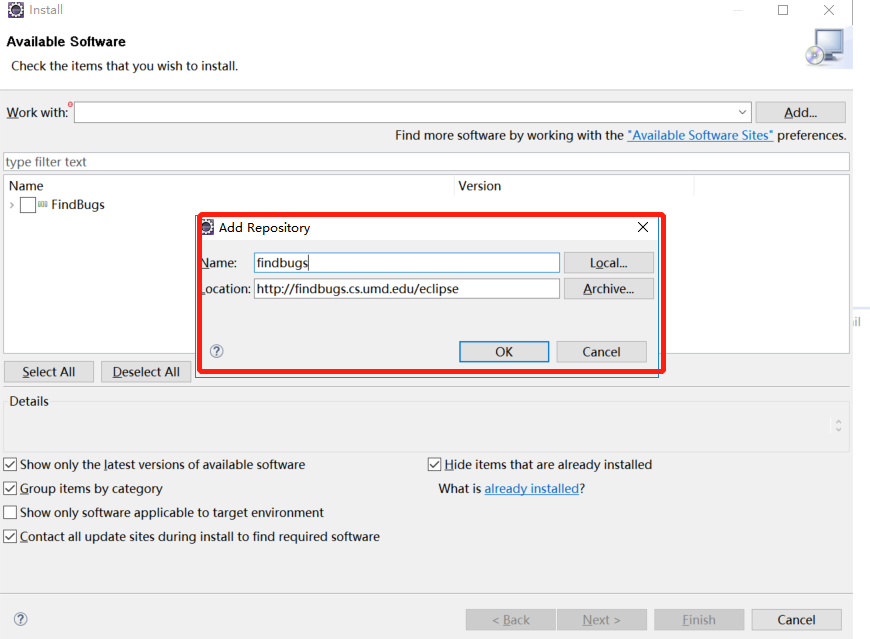
|  |
| --- |
| 黑色的臭虫标志是分类；  红色的臭虫标志表示严重bug，发现后必须修改代码；  橘黄色的臭虫标志表示潜在警告性bug，应尽量修改； |

## Eclipse插件安装

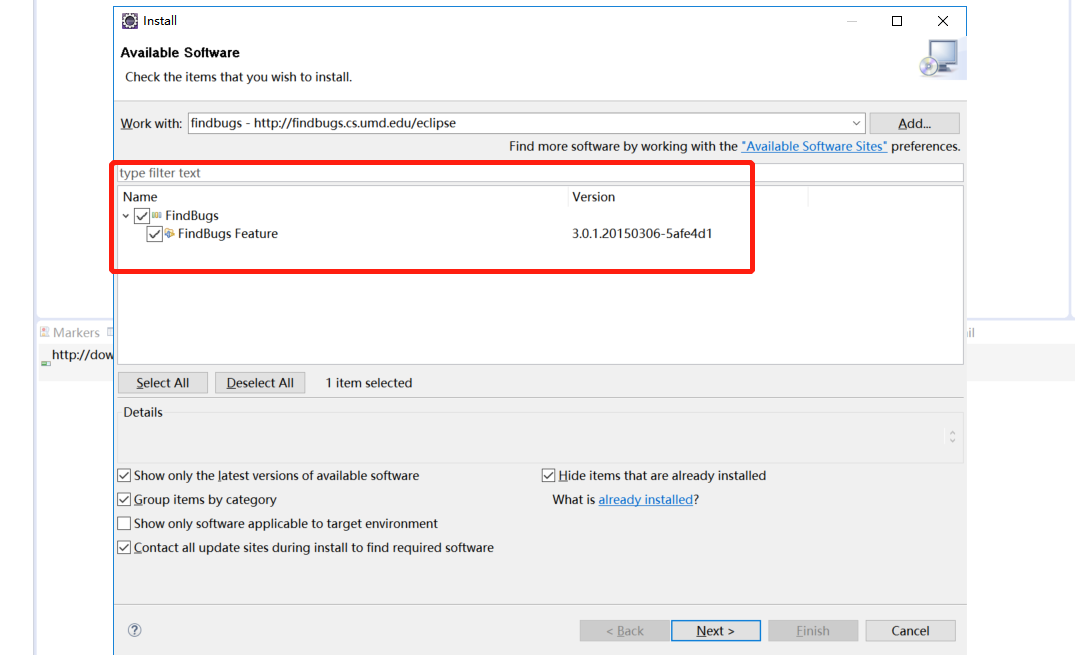
### 在线安装

1 点击“Help->InstallNew Software”

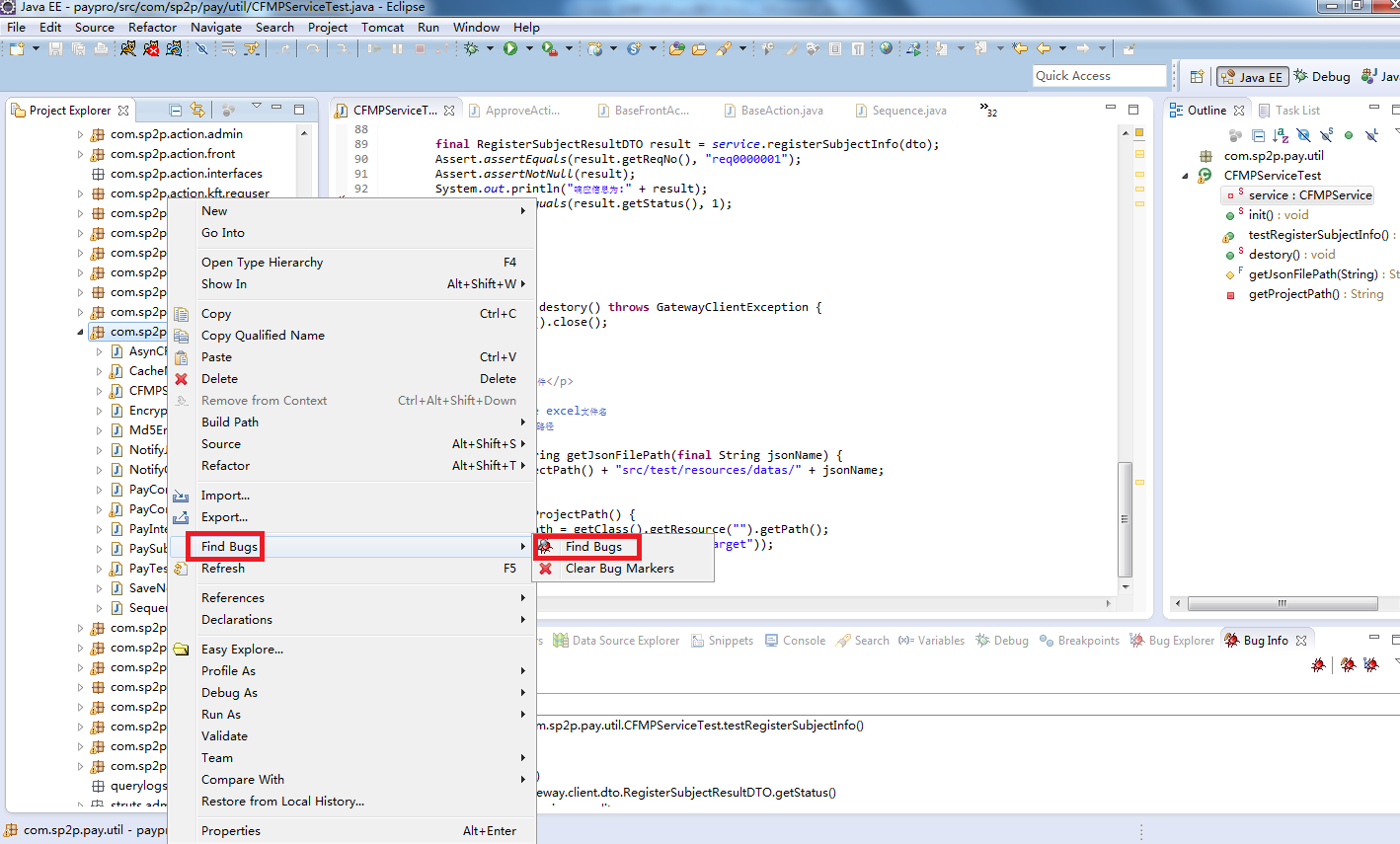
2点击“Add”，然后在弹出框“Name”输入“findBugs”，“Location”输入“<http://findbugs.cs.umd.edu/eclipse>”，点击“OK”，如下图：



3 选择对应插件，然后点击“next->next->finish”



4完成安装之后重启eclipse，右击项目文件或目录，会发现多了Findbugs的菜单，如下图：



### 离线安装

安装包下载地址: [http://findbugs.sourceforge.net/downloads.html 本文档使用如下](http://findbugs.sourceforge.net/downloads.html，本文档使用如下)



然后将文件复制到你本地eclipse的plugins目录，然后重启eclipse即可。

### findBugs配置

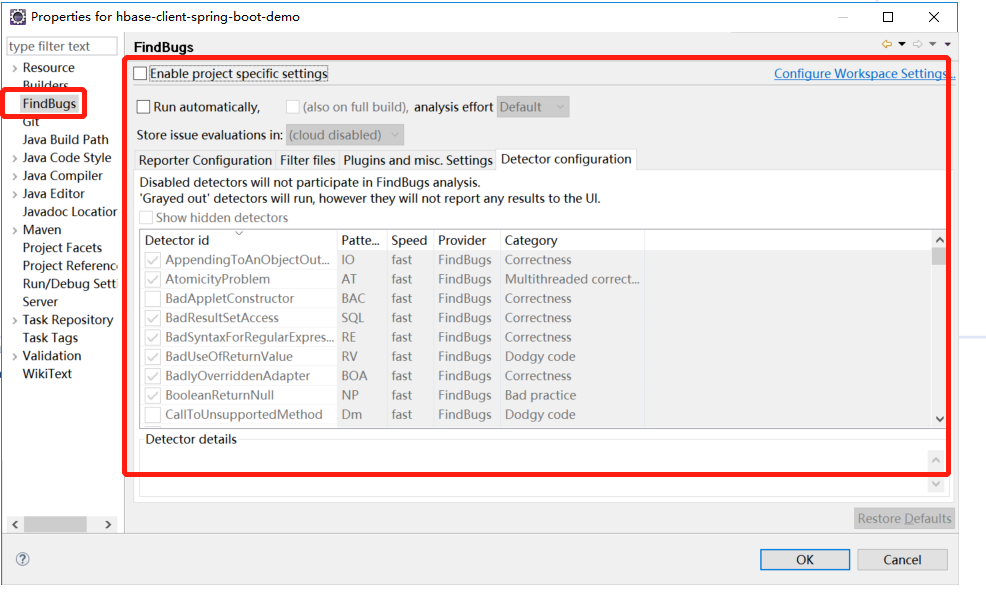
在这里可以对FindBugs规则等进行详细设置。

选择你的项目，右键 => Properties => FindBugs =>

1 Run Automatically开关

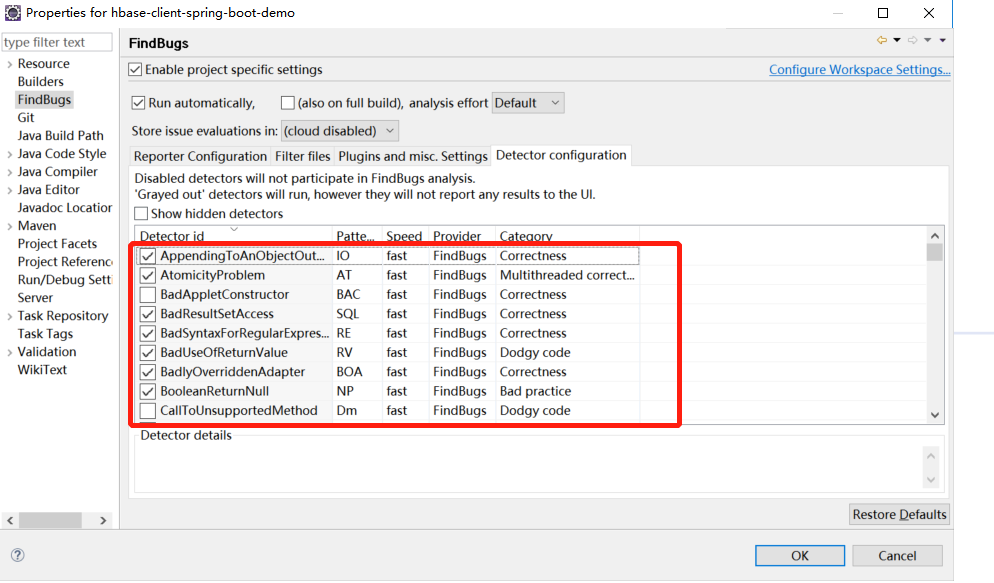
当此项选中后，FindBugs将会在你修改Java类时自动运行，如你设置了Eclipse自动编译开关后，当你修改完Java文件保存，FindBugs就会运行，并将相应的信息显示出来。

当此项没有选中，你只能每次在需要的时候自己去运行FindBugs来检查你的代码。



2 Detector Configuration选择项

在这里你可以选择所要进行检查的相关的Bug Pattern条目,你可以根据需要选择或去掉相应的 检查条件。



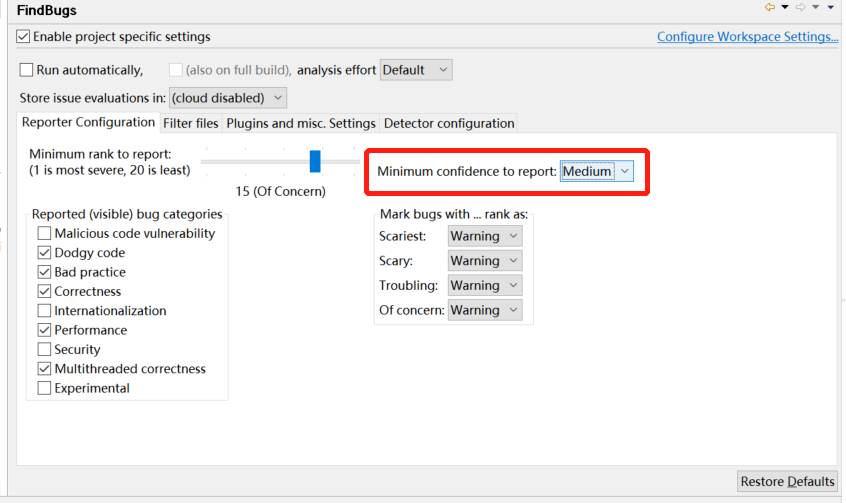
3 Minimum priority to report选择项

这个选择项是让你选择哪个级别的信息进行显示，有Low、Medium、High三个选择项可以选择，很类似于Log4J的级别设置啦。 比如：

你选择了High选择项，那么只有是High级别的提示信息才会被显示。

你选择了Medium选择项，那么只有是Medium和High级别的提示信息才会被显示。

你选择了Low选择项，那么所有级别的提示信息都会被显示。



4 Report bug categories选择项

在这里是一些显示Bug分类的选择：

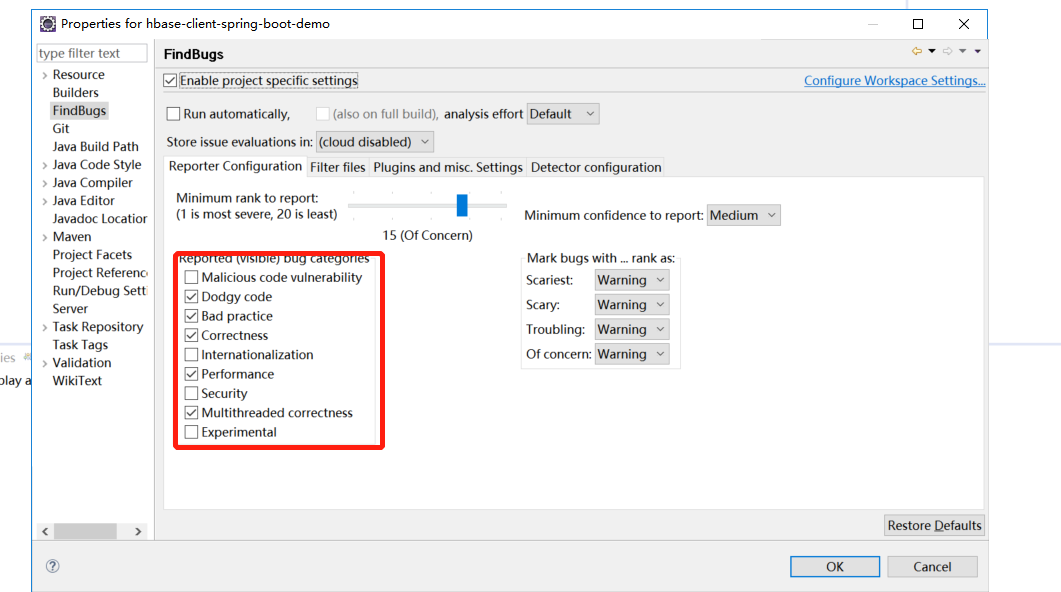
Malicious code vulnerability 关于恶意破坏代码相关方面的

Correctness 关于代码正确性相关方面的

Internationalization 关于代码国际化相关方面的

Performance 关于代码性能相关方面的

Multithreaded correctness关于代码多线程正确性相关方面的



大的分类主要包括以下几种：

|  |  |
| --- | --- |
| Bad practice | 不好的习惯 |
| Correctness | 代码的正确性 |
| Dodgy | 小问题 |
| Malicious code vulnerability | 恶意代码 |
| Internationalization | 国际化问题 |
| Performance | 性能问题 |
| Security | 安全性问题 |
| Multithreaded currectness | 线程问题 |
| Experrimental | 实验性问题 |

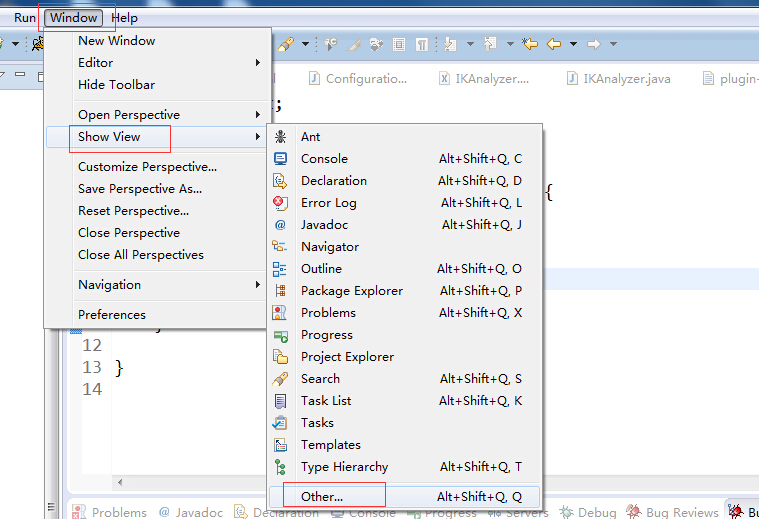
### findBugs使用

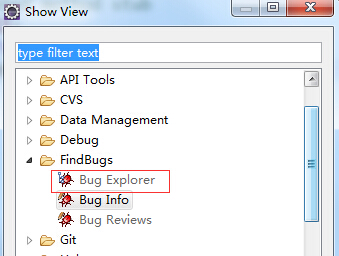
直接在要扫描的文件或项目下点击右键，选择findbugs即可

1. 安装完成并重启后，选中一个项目，右键如图，选择Find Bugs：

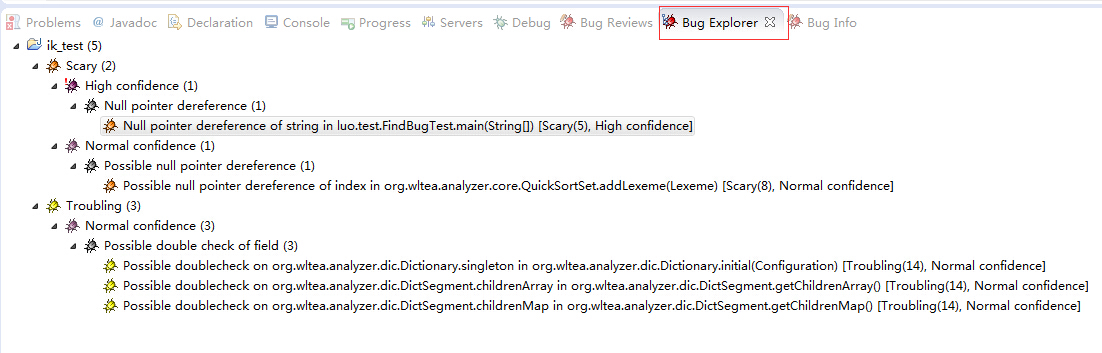


1. 运行完成后查看Bug Explorer视图，如果没有该视图，操作如下图:

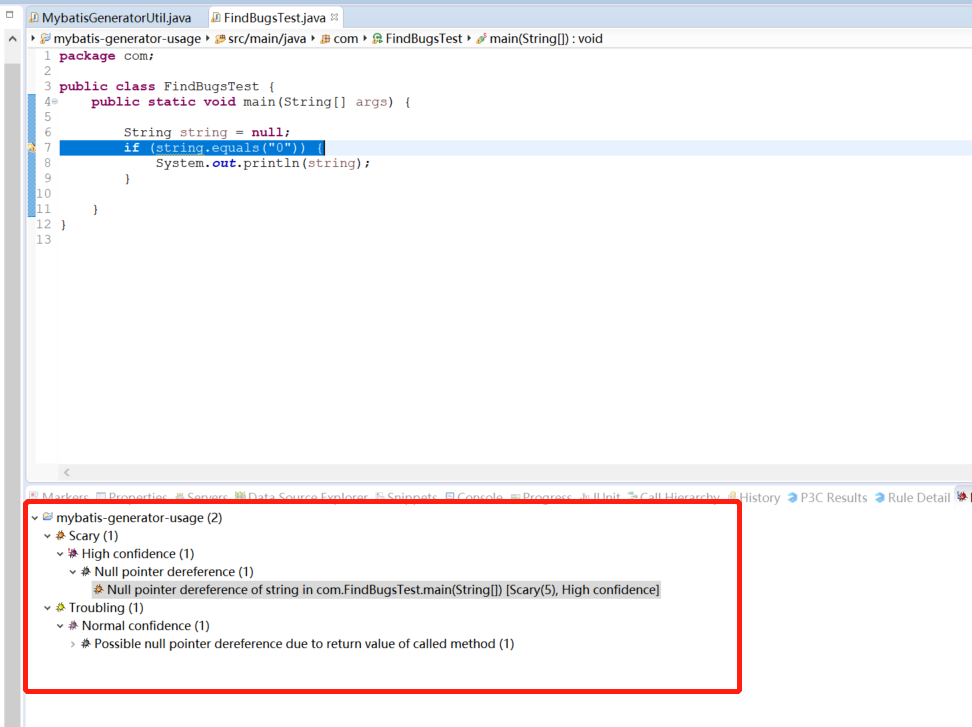




1. 如果检查出你的代码问题会出现如下图



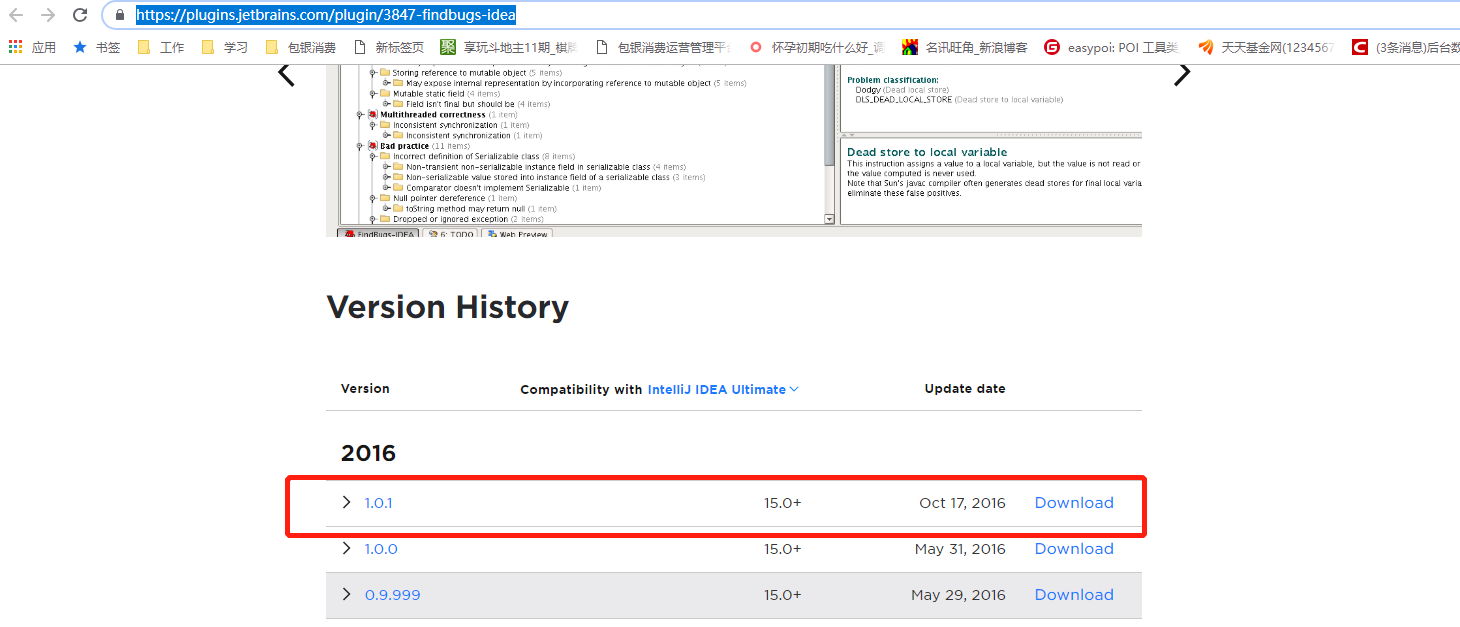
1. 点击其中一个叶子节点



## Idea插件安装

### 下载

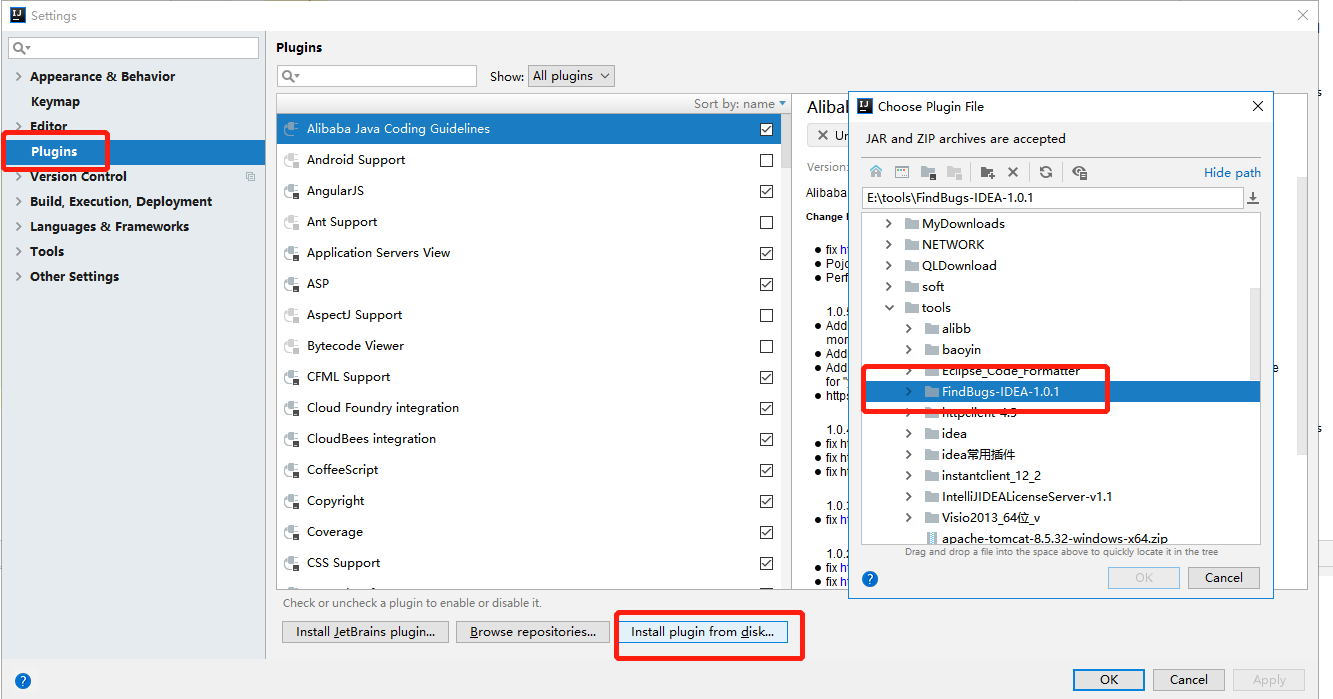
下载地址:https://plugins.jetbrains.com/plugin/3847-findbugs-idea



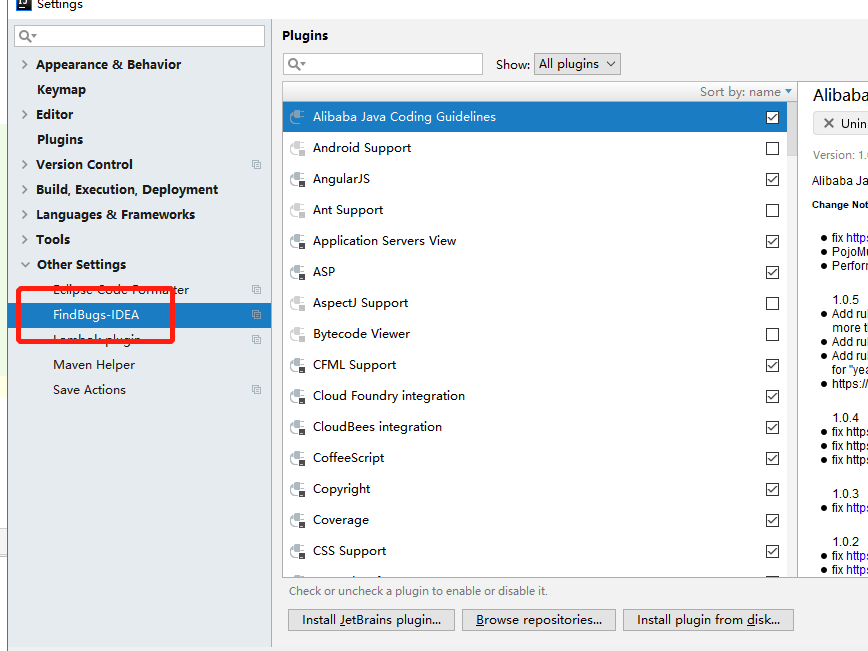


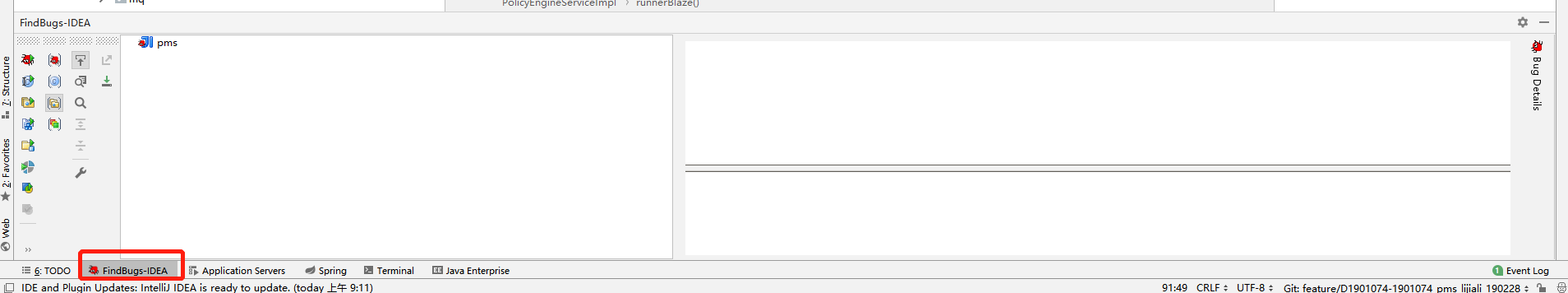
### findBugs安装

安装步骤：File--->setting--->Plugins--->Install......--->选择 6.3.1中下载的findBugs包



安装重启后，在Tools下会显示如图信息:

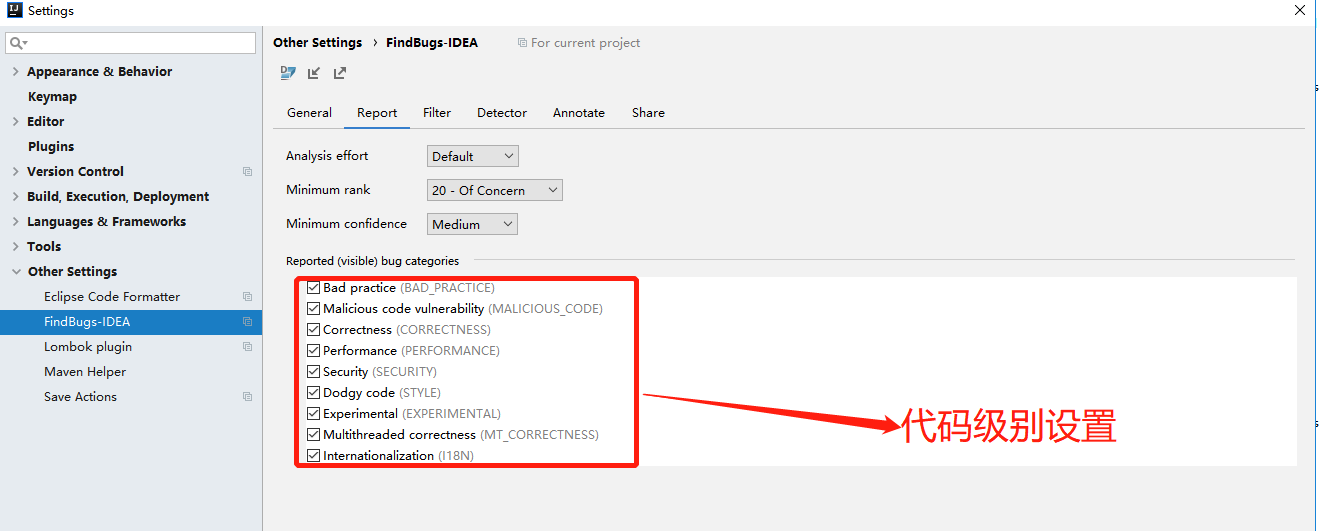




### findBugs配置

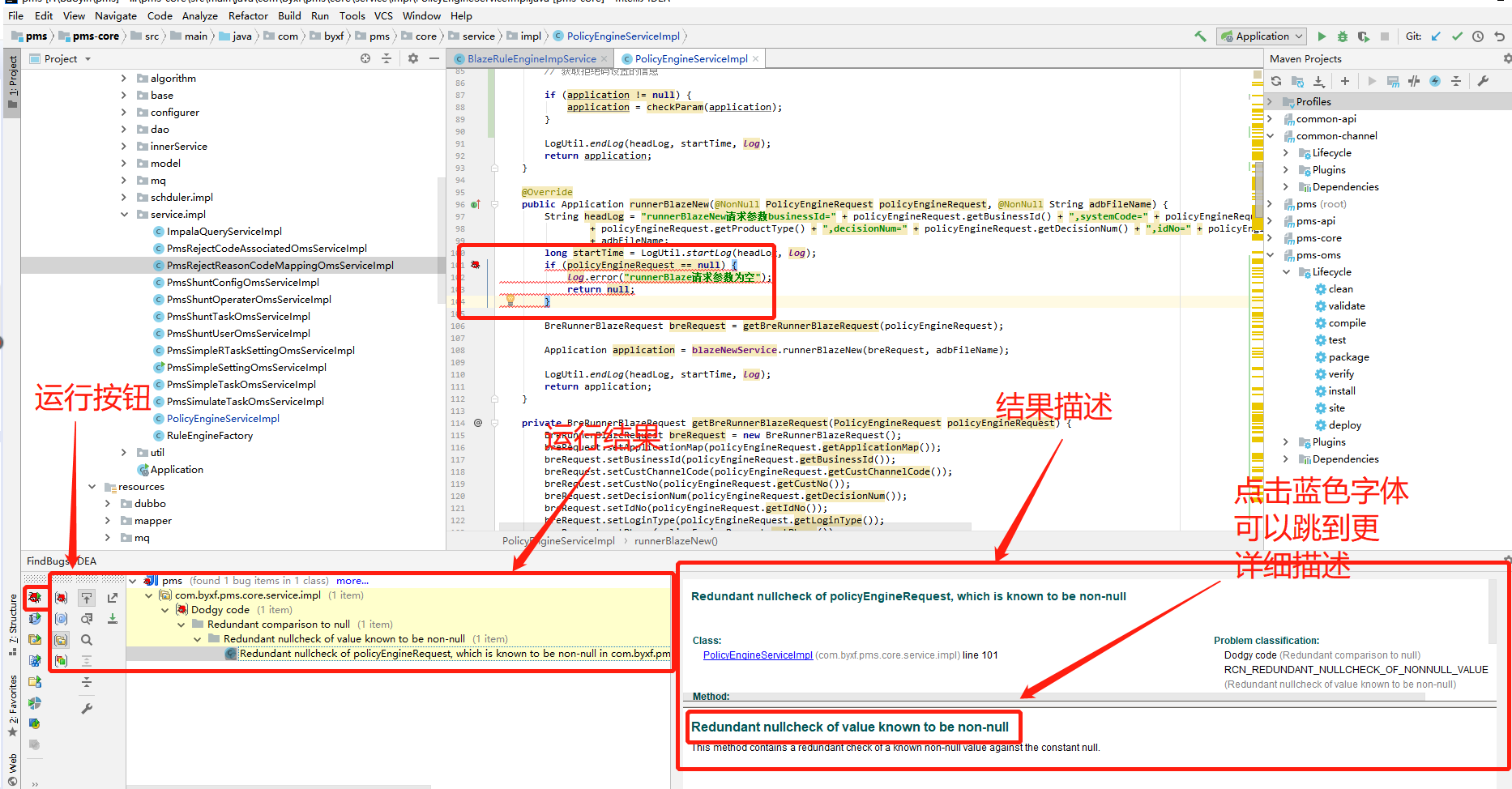
可以根据需要设置相应的扫描级别

|  |  |
| --- | --- |
| Bad practice | 不好的习惯 |
| Correctness | 代码的正确性 |
| Dodgy | 小问题 |
| Malicious code vulnerability | 恶意代码 |
| Internationalization | 国际化问题 |
| Performance | 性能问题 |
| Security | 安全性问题 |
| Multithreaded currectness | 线程问题 |
| Experrimental | 实验性问题 |



### findBugs使用

单文件扫描:打开要扫描的文件，点击左下角的运行按钮即可,截图如下:

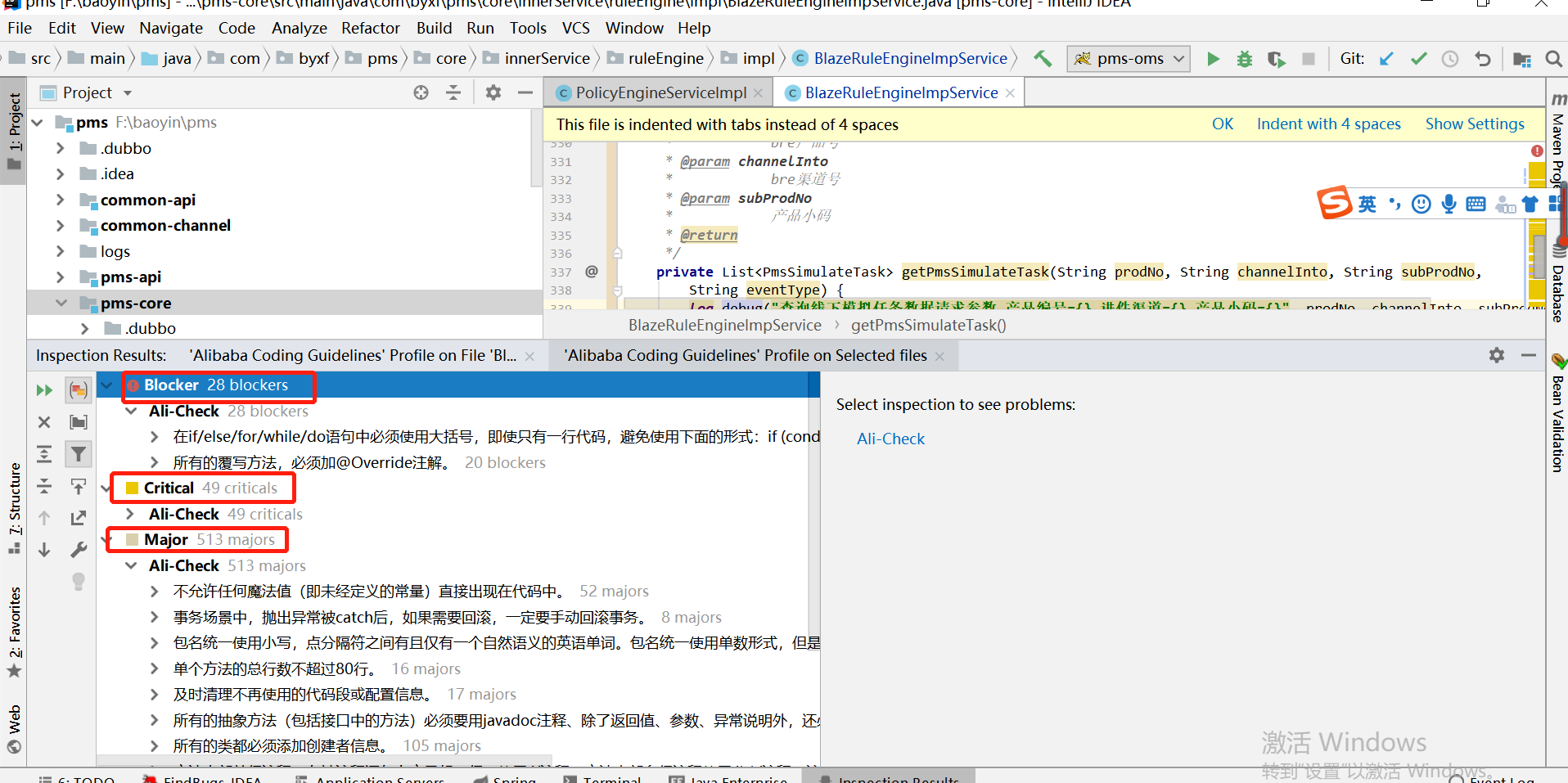




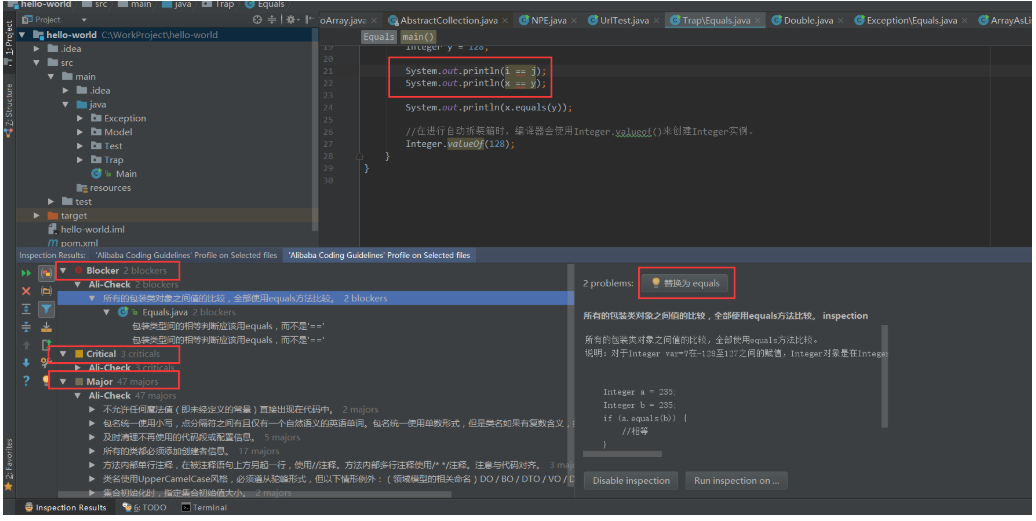
# 扫描结果分析

## 阿里编码规约

扫描代码后，不符合规约的代码会按Blocker/Critical/Major三个等级显示在下方面板中，解释下Blocker/Critical/Major三个等级，在 Snoar 中对代码规则有五个级别，这是前三个：崩溃/严重/重要 ，也就是说前两级别是必须要处理掉的。如图:



左边是扫描出的不符合规范的代码，依次点进去可以看到是代码的多少行出现了规约问题以及哪一个规约问题，右边则是规约的详细描述及实例代码。



不仅如此，右侧还有quick fix的按钮，点击后直接可以改正代码，但是并不是所有的问题都有此按钮，有些问题还是需要手动修改。

## findBugs扫描结果

# 总结

这个插件可以说对规范代码有重大意义，可以使我们少写一些 Bug，少抛一些异常，少踩一些坑，但真正要写出健壮代码最重要的还在于我们自己，目前此插件还处于第一个公开版本，所以肯定还有一些小问题或者小bug，以后应该会慢慢修复的，但是已经基本满足代码审查的需求了，用起来也比较满意。不过，一些容易产生空指针异常的语句和问题，好像并不能检查出来，只是满足基本的代码质量检测。