目录

[idea 简介 1](#_Toc491811530)

[创建项目 3](#_Toc491811531)

[jar项目使用方式 4](#_Toc491811532)

[java web项目 6](#_Toc491811533)

[创建maven项目 8](#_Toc491811534)

[导入项目 9](#_Toc491811535)

[idea 代码调试 11](#_Toc491811536)

[java代码调试 11](#_Toc491811537)

[idea 高级应用 13](#_Toc491811538)

1.什么是Idea？

2.为什么要使用这个？

3.怎么使用?

# idea 简介

IDEA 全称 IntelliJ IDEA，是java语言开发的集成环境，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、各类版本工具(git、svn、github等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。IDEA是[JetBrains](https://baike.baidu.com/item/JetBrains" \t "_blank)公司的产品，这家公司总部位于[捷克共和国](https://baike.baidu.com/item/%E6%8D%B7%E5%85%8B%E5%85%B1%E5%92%8C%E5%9B%BD" \t "_blank)的首都[布拉格](https://baike.baidu.com/item/%E5%B8%83%E6%8B%89%E6%A0%BC" \t "_blank)，开发人员以严谨著称的东欧程序员为主。它的旗舰版本还支持HTML，CSS，PHP，MySQL，Python等

总结：idea自动化的功能比较全的一个IDE工具

# IDEA优点

　1. 智能的选取。在很多时候我们要选取某个方法，或某个循环或想一步一步从一个变量到整个类慢慢扩充着选取，IDEA就提供这种基于语法的选择，在默认设置中Ctrl+w，可以实现选取范围的不断扩充，这种方式在重构的时候尤其显得方便。

　　2. 丰富的导航模式。IDEA提供了丰富的导航查看模式，例如ctrl+e显示最近打开过的文件，ctrl+n显示你希望显示的类名查找框

　　(该框同样有智能补充功能，当你输入字母后IDEA将显示所有候选类名)。在最基本的project试图中，你还可以选择多种的试图方式。

　　3. 历史记录功能。不用通过版本管理服务器，单纯的IDEA就可以查看任何工程中文件的历史记录，但版本恢复时你可以很容易的将其恢复。

　　4. JUnit的完美支持。

　　5. 对重构的优越支持。IDEA是所有IDE中最早支持重构的，其优秀的重构能力一直是其主要卖点之一。

　　6. 编码辅助。java规范中提倡的tostring(),hashcode(),equals(),以及所有的get/set方法，你可以不用进行任何的输入就可以实现代码的自动生成，从而把你从无聊的基本方法编码中解放出来。

　　7. 灵活的排版功能。基本所有的IDE都有重排版功能，但仅有IDEA的是人性的，因为她支持排版模式的定制，你可以根据不同的项目要求采用不同的排版方式。

　　8. XML的完美支持。

　　9. 简洁易用的GUI界面。

　　10. 与文件系统自动同步。当你最小化IDEA，直接对文件进行编辑之后，重新打开IDEA后IDEA将自动的完成文件的更新，而不用重新导入工程或刷新工程。

　　11. 自定义的快捷键。强大的快捷键设置，基本可以让程序员离开鼠标工作。

　　12. 动态语法检测。任何不符合java规范，自己预定义的规范，累赘都将在页面中加亮显示。

　　13. 代码检查。对代码进行自动分析，检测不符合规范的，，存在风险的代码，并加亮显示。

　　14. 对JSP的完全支持。不需要任何的插件，完全支持JSP。

　　15. 智能编辑。代码输入过程中，自动补充方法或类。

　　16. EJB支持。不需要任何插件完全支持EJB(6.0 支持ejb3.0)

　　17. 列编辑模式。用过UtralEdit的肯定对其的列编辑模式赞赏不已，因为她减少了很多很多的无聊重复工作，而IDEA完全支持该模式，从而更加提高了编码效率。

　　18. 智能模板。预置模板可以让你把经常用到的方法编辑进模板，使用时你只用输入简单的几个字母就可以完成全部代码的编写。例如使用比较高的public static void main(String[] args){}你可以在模板中预设pm为该方法，输入时你只要输入pm再按代码辅助键，IDEA将完成代码的自动输入。

　　19. 完美的自动代码完成。智能检查类中方法，当发现方法名只有一个时自动完成代码输入，从而减少剩下代码的编写工作。

　　20. Ant 支持。不需要任何的第三方插件。

　　21. 不使用代码检查。自动检查代码中不使用的代码，并给出提示，从而使代码更高效。

　　22. 智能代码。自动检查代码，发现与预置规范有出入的代码给出提示，若程序员同意修改自动完成修改。例如代码：String str = "Hello Intellij " + "IDEA"; IDEA将给出优化提示，若程序员同意修改IDEA将自动将代码修改为：String str = "Hello Intellij IDEA";

　　23. 正则表达的查找和替换功能。查找替代支持正则表达，从而提高效率。

　　24. JavaDoc预览支持。支持JavaDoc的预览功能，在JavaDoc代码中ctrl+q显示JavaDoc的结果，从而提高doc文档的质量。

25. 程序员意图支持。当程序员编码时IDEA时时检测你的意图，或提供建议，或直接帮你完成代码

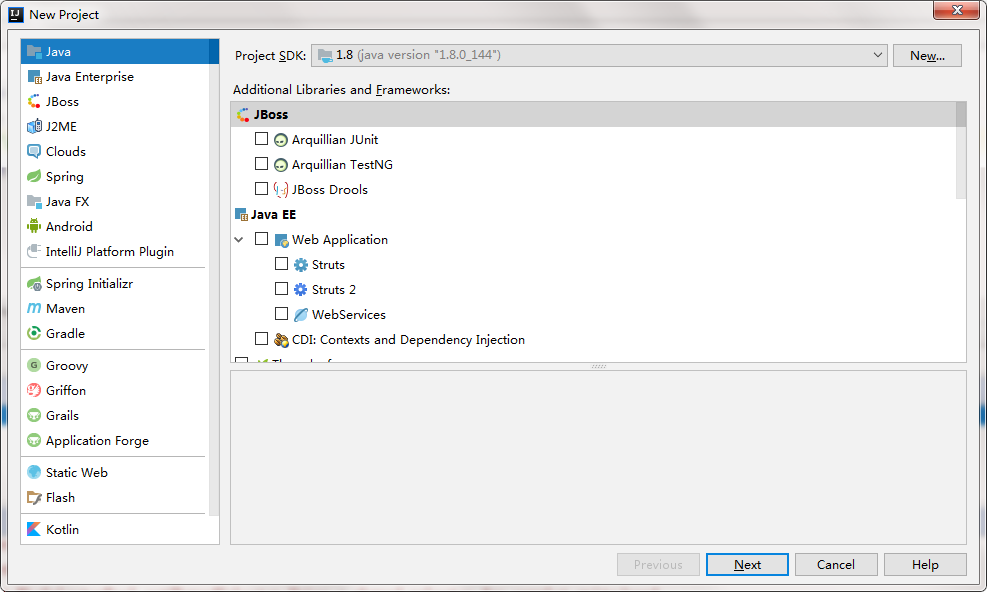
总结：快捷，搜索，分析，定位，重构，高效。

# 创建项目

## jar项目使用方式

新建项目：

File –》 new --》 Projects..

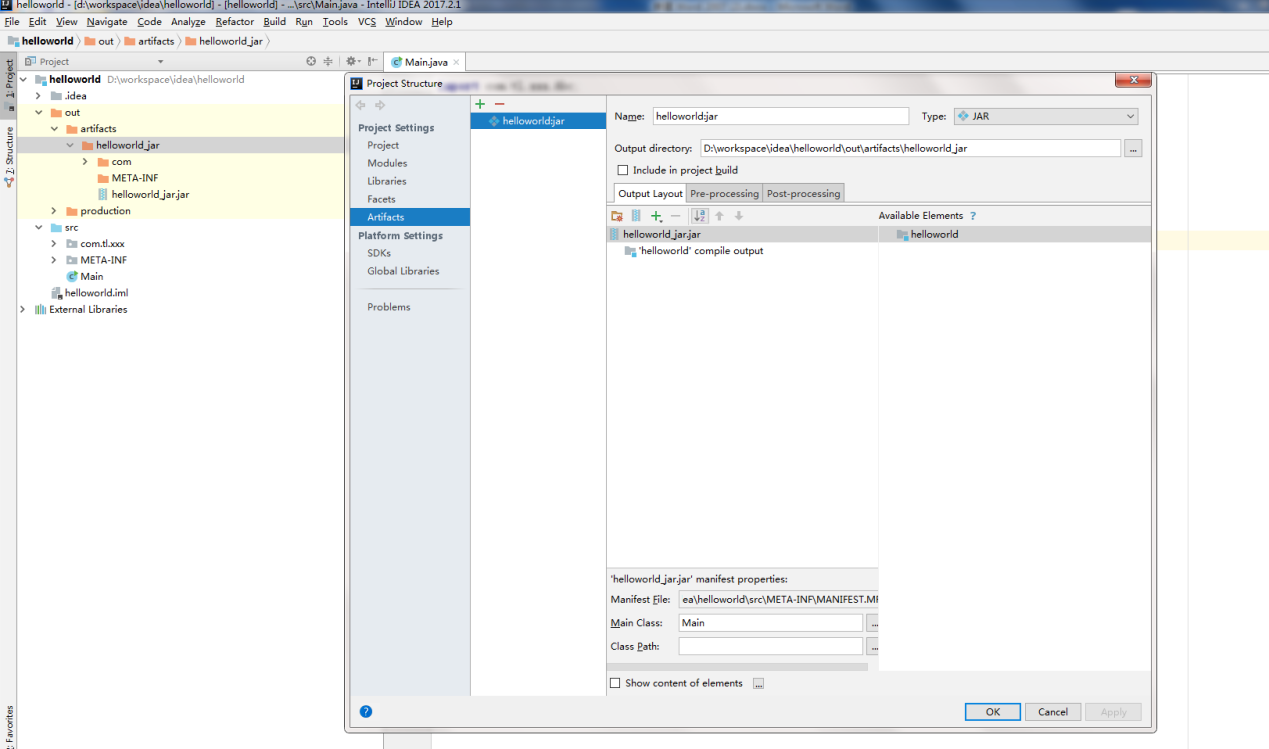


打包配置方式

File –》Project Structure –》Artifacts 点击

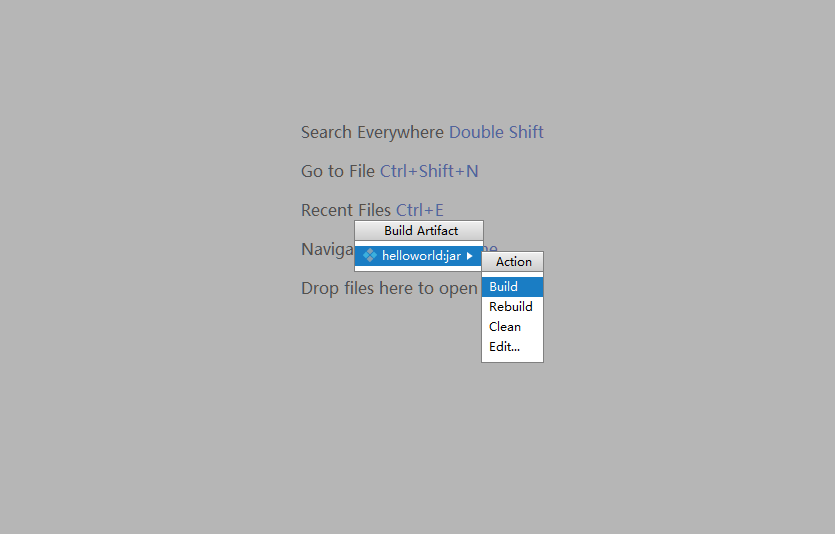
🡪 ＋号 –》 选择 jar --》 From modules with dependencies...

选择Main class 启动项目的主类。



打包输出方式：

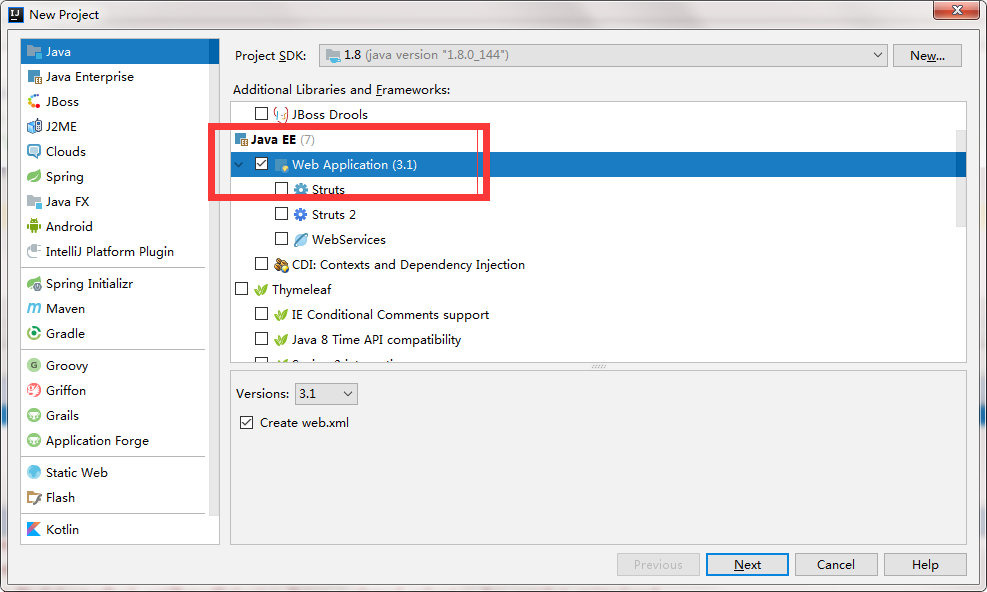
Build 🡪 Build Artifacts 选择具体的jar



## java web项目

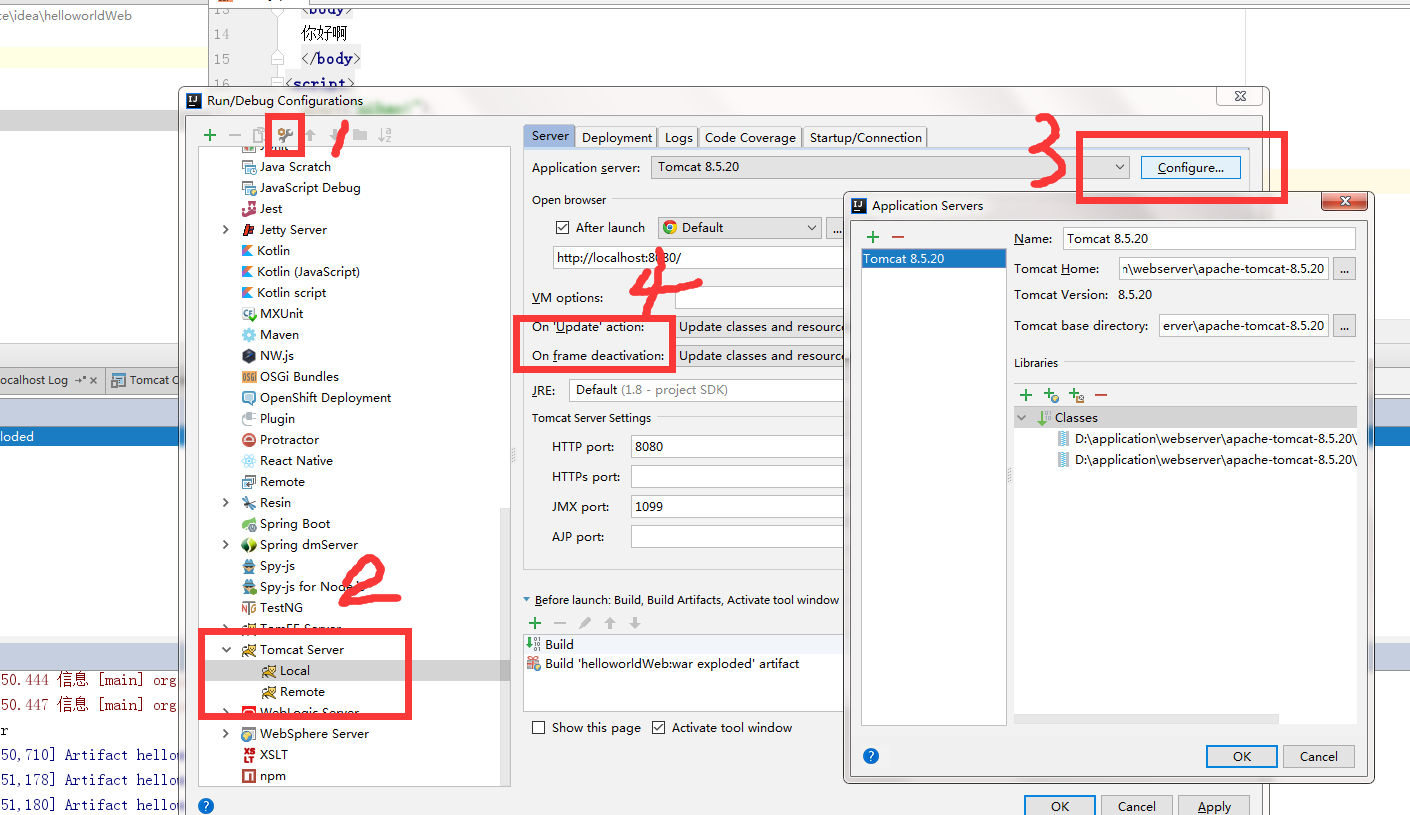
创建项目：

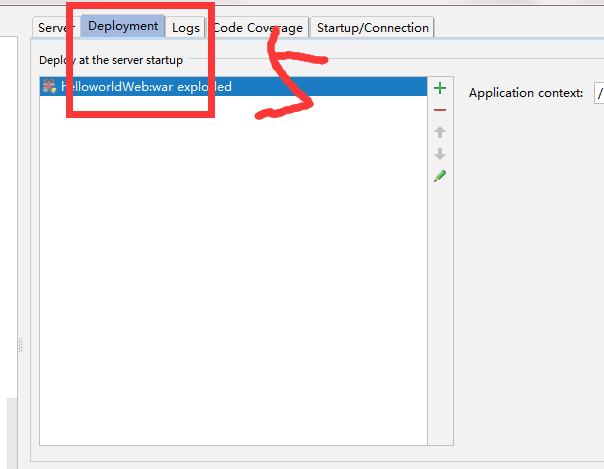
File –》 new --》 Projects..

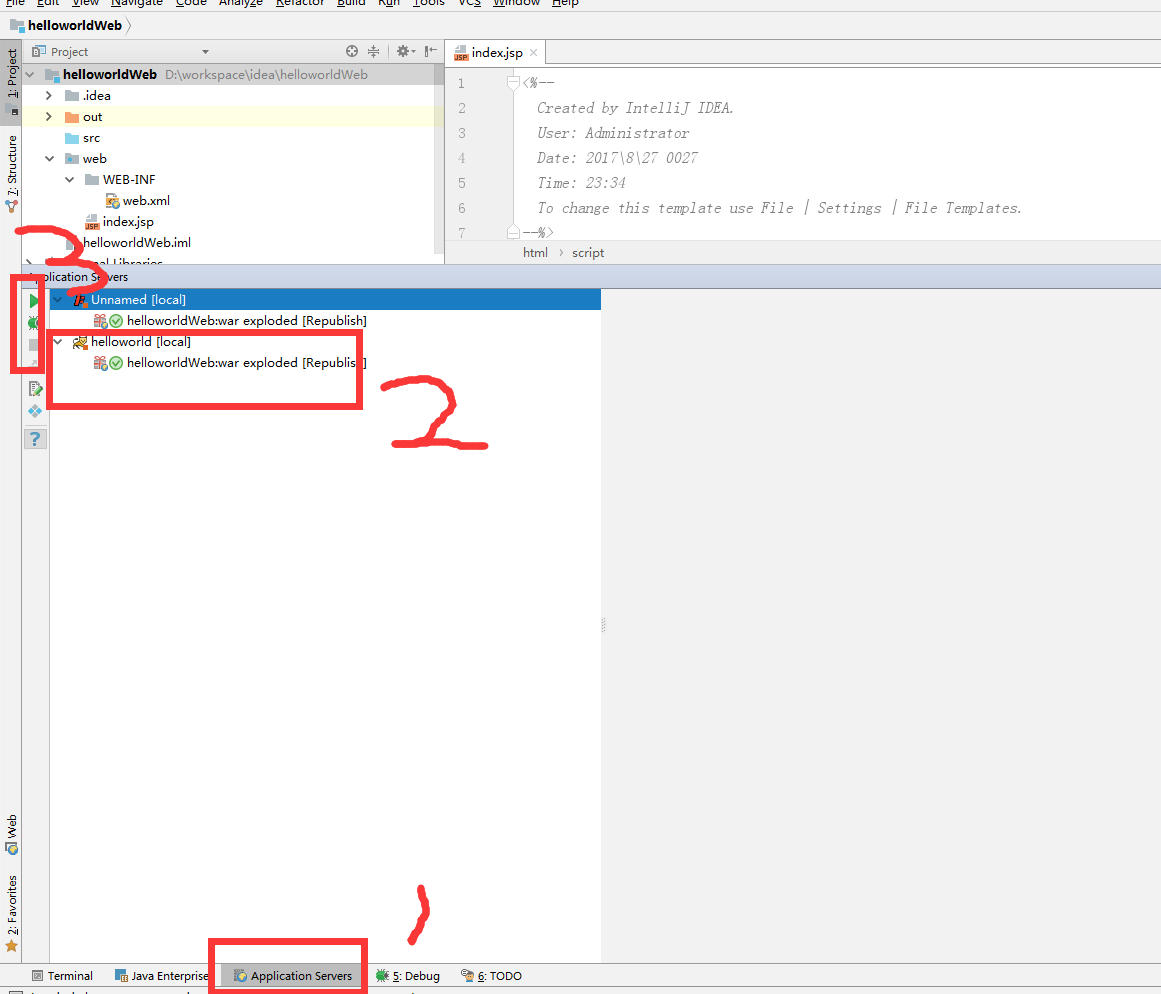


配置web服务器启动

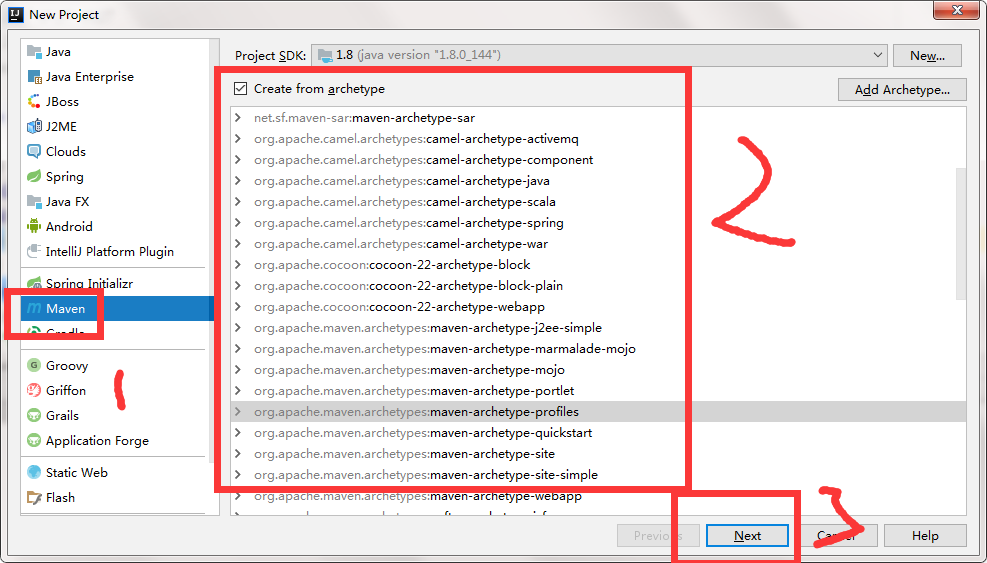
Edit Configurations...

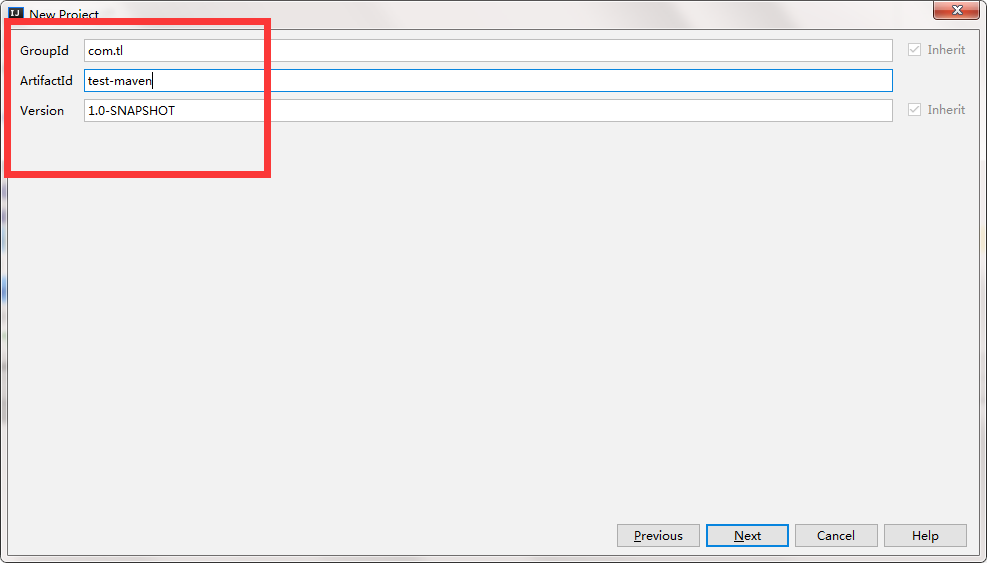






## 创建maven项目

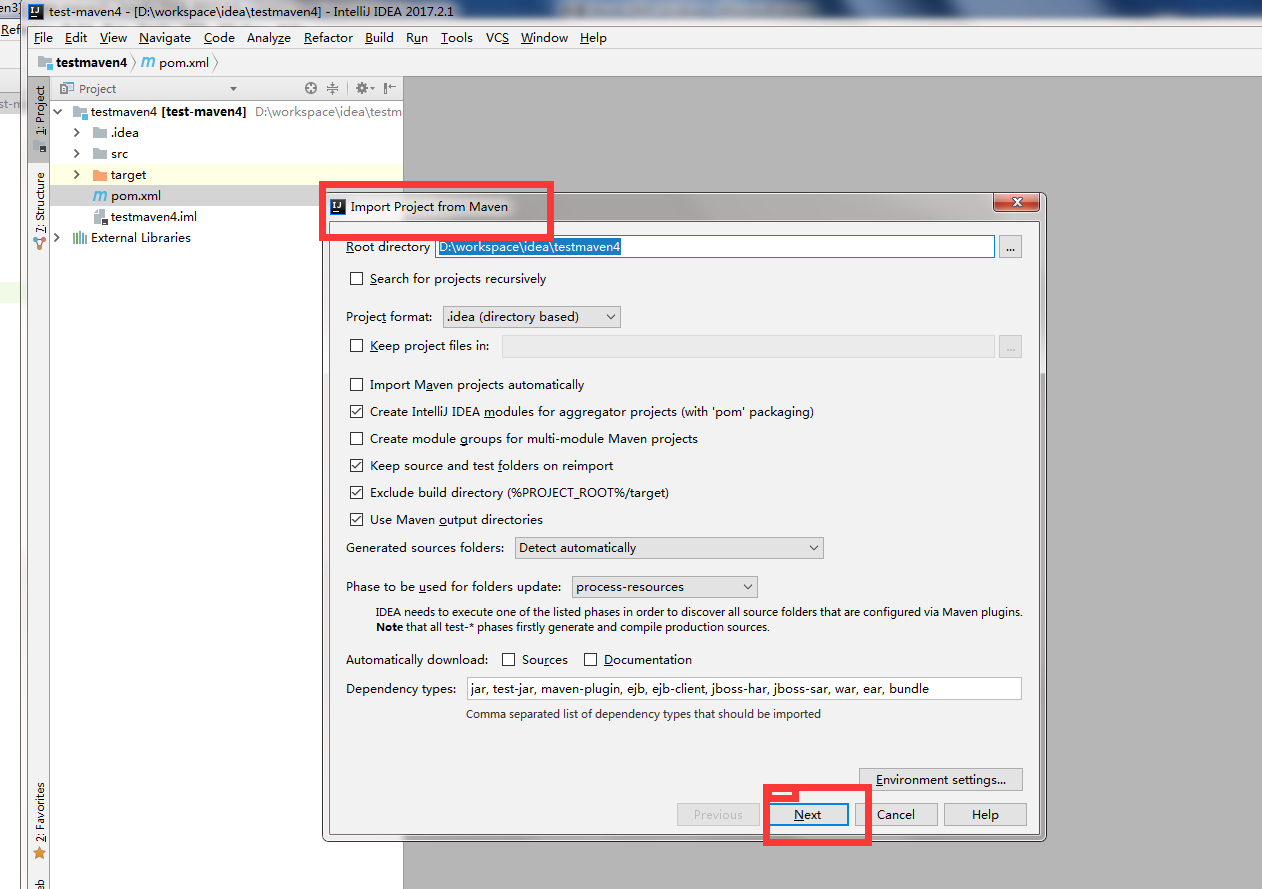




## 导入项目

File –》 new –》Project from existing Sources

然后一路 Next 即可

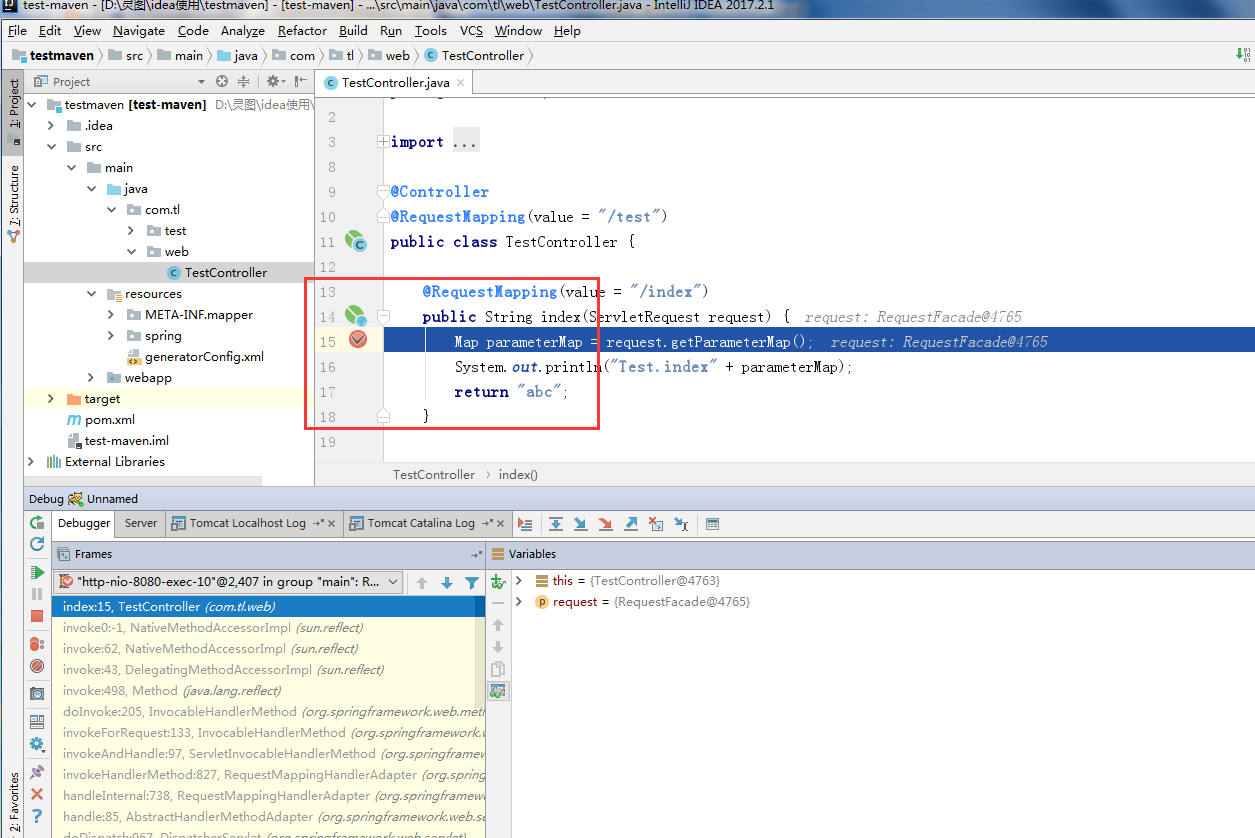


# idea 代码调试

## java代码调试

本地调试

代码中直接执行加断点。



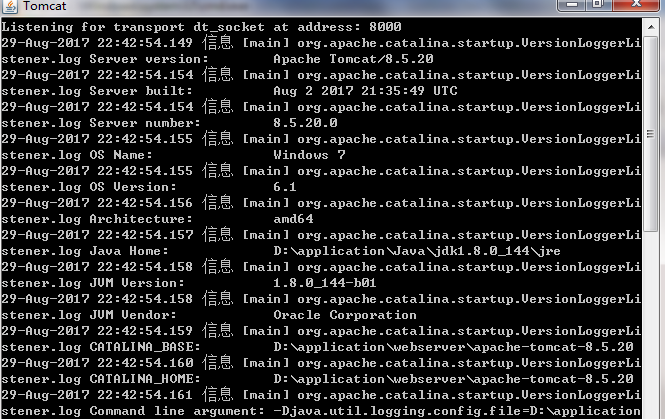
远程调试测试

把测试项目放入到tomcat的webapp目录

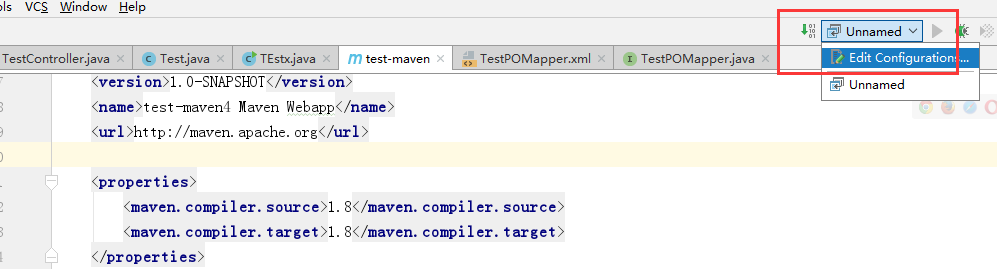
tomcat8 开启远程调试:

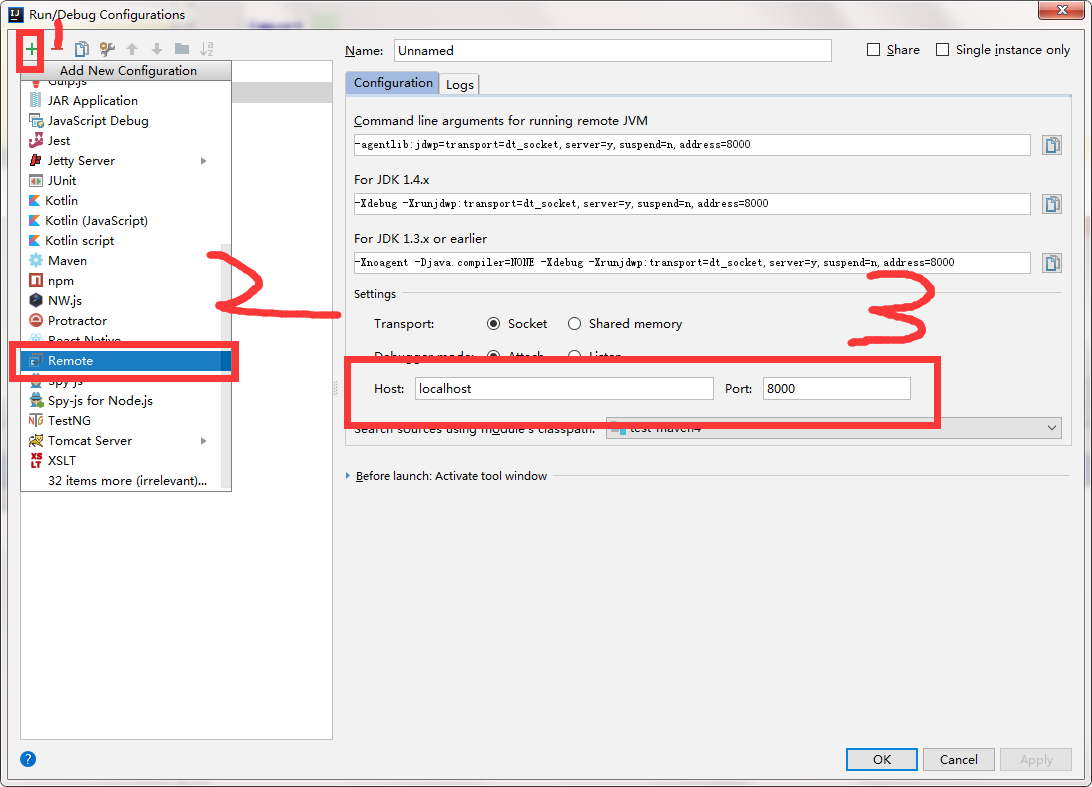
catalina.bat jpda start

显示listening xxx 端口就证明开启了远程调试模式了



Edit Configurations...



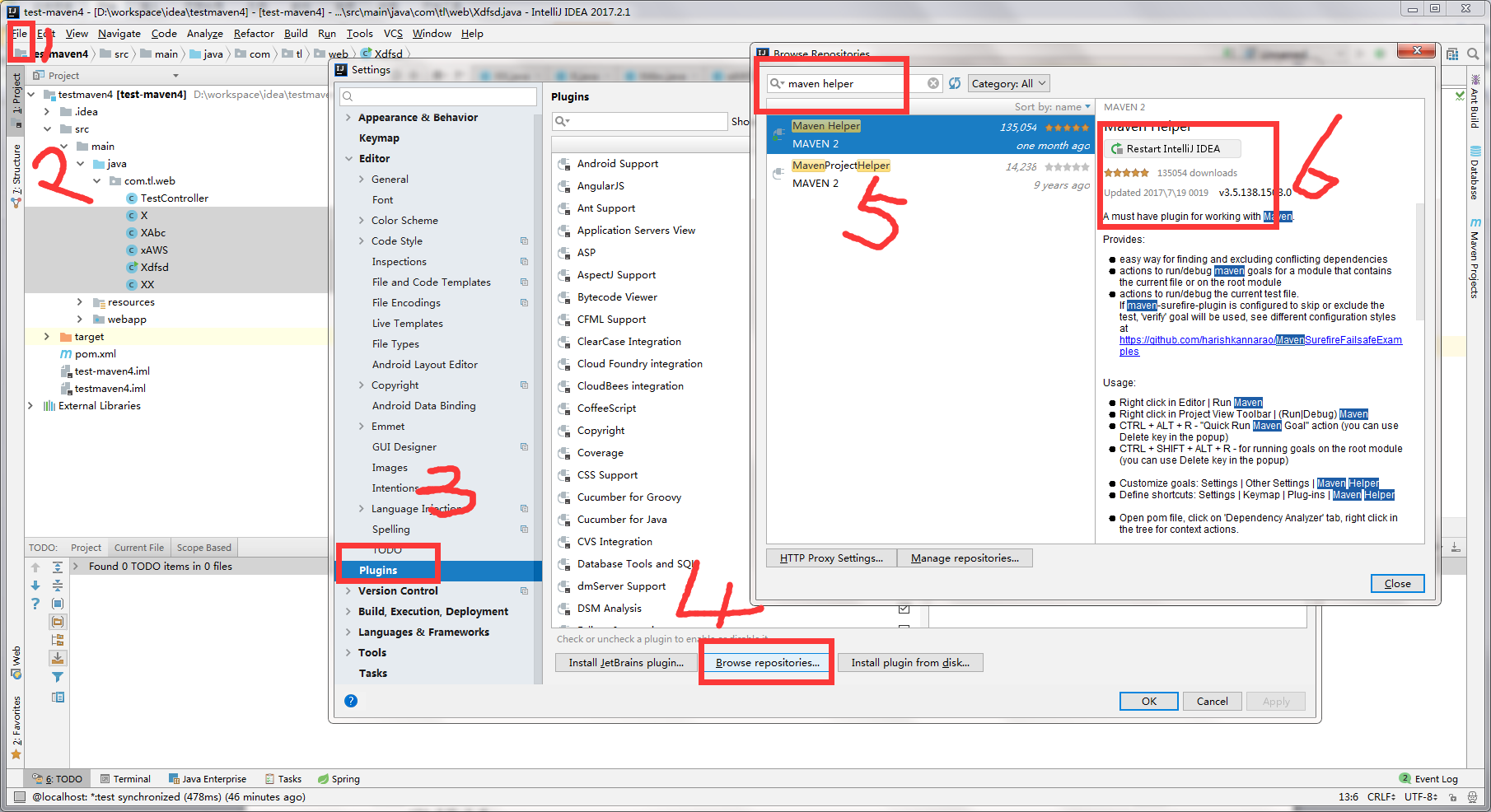


其他的调试：js，其他语言的调试等等

# idea 高级应用

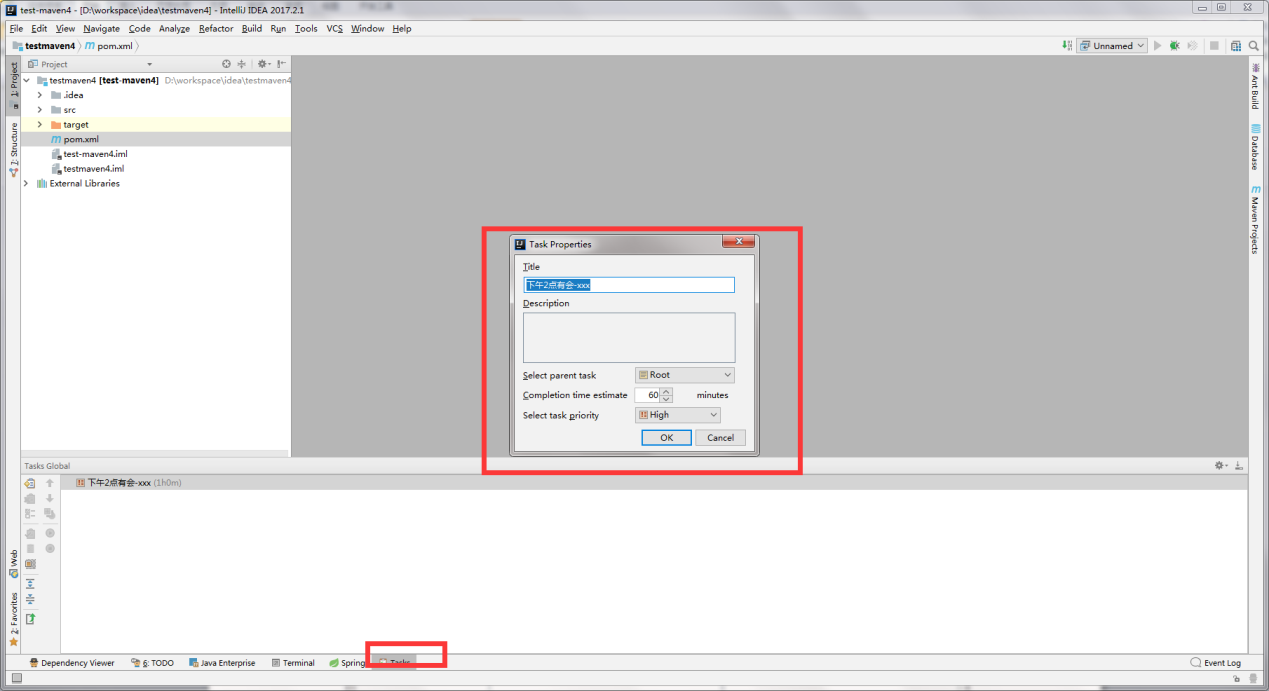
## 1.插件安装使用

安装两种方式一种是远程，一种本地

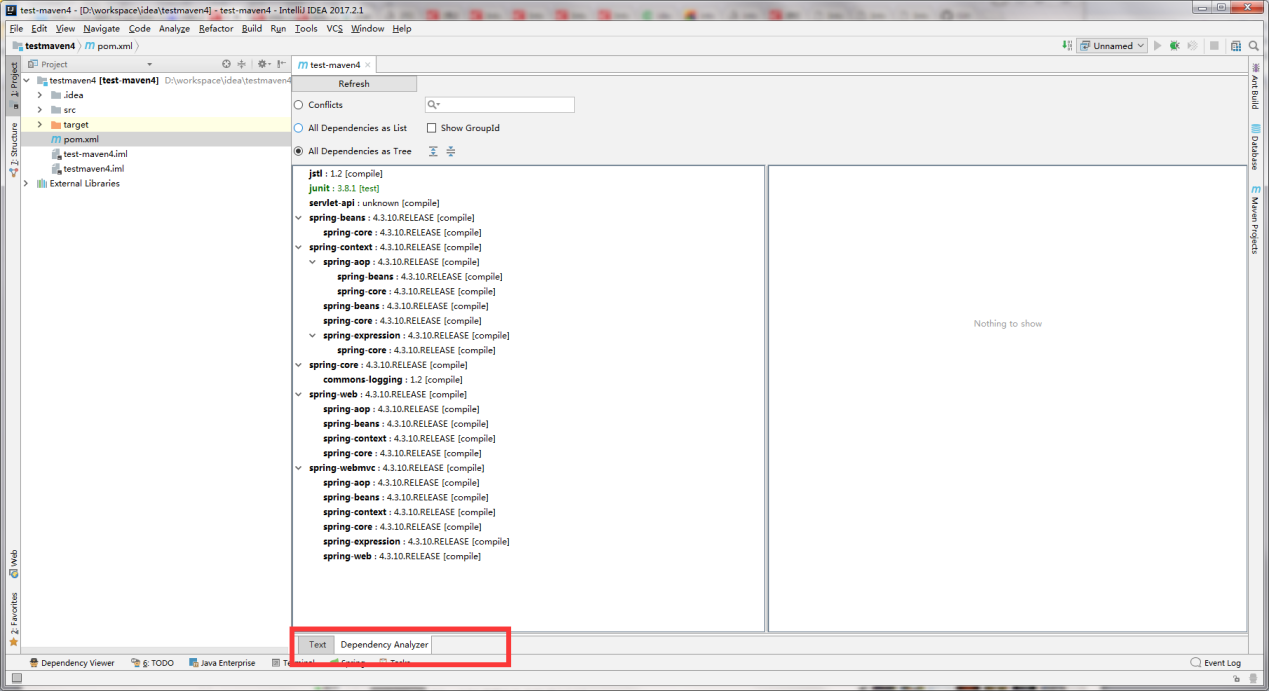


mybatis插件使用

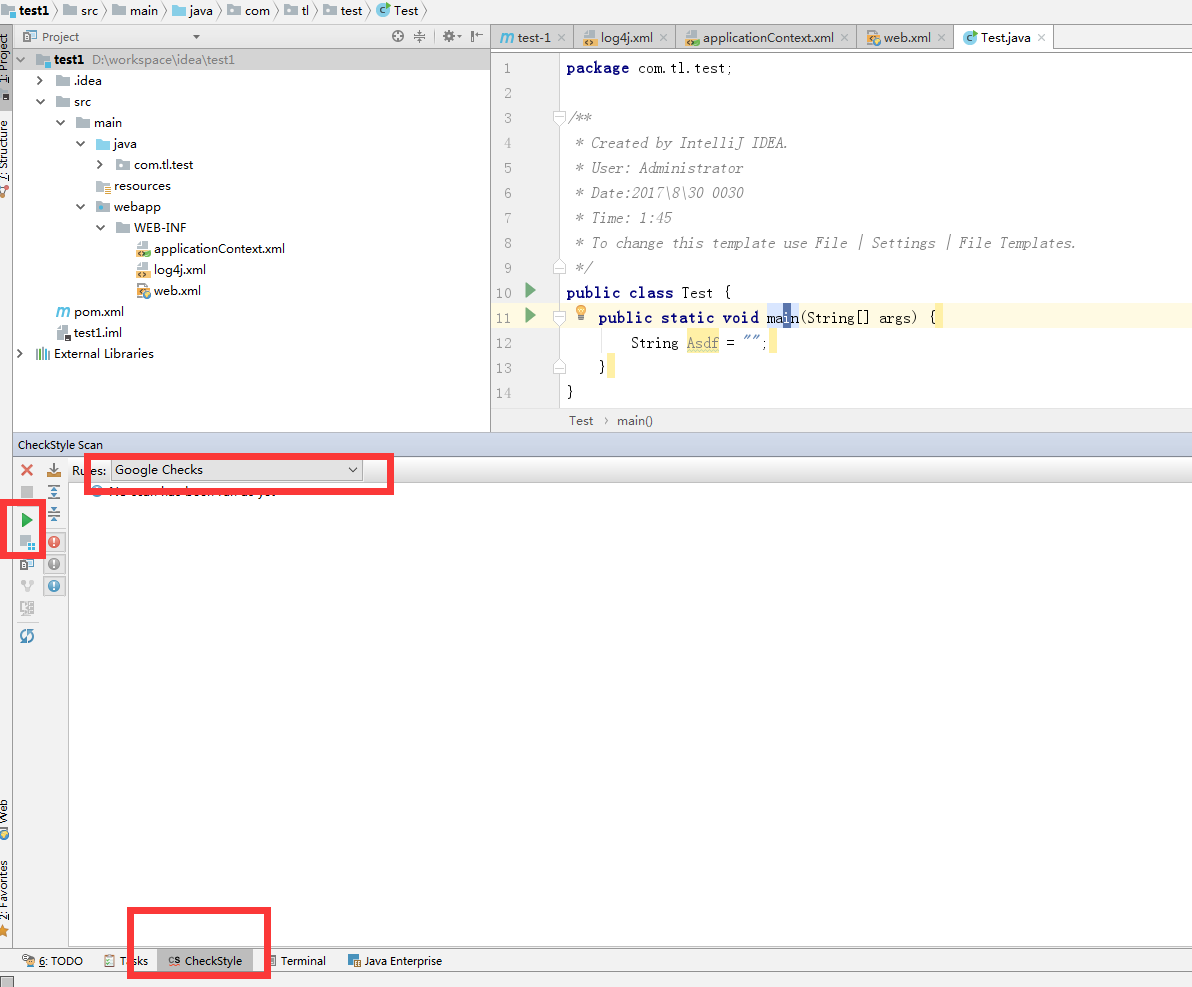
tasks插件



maven-helper 插件



checkstyle使用



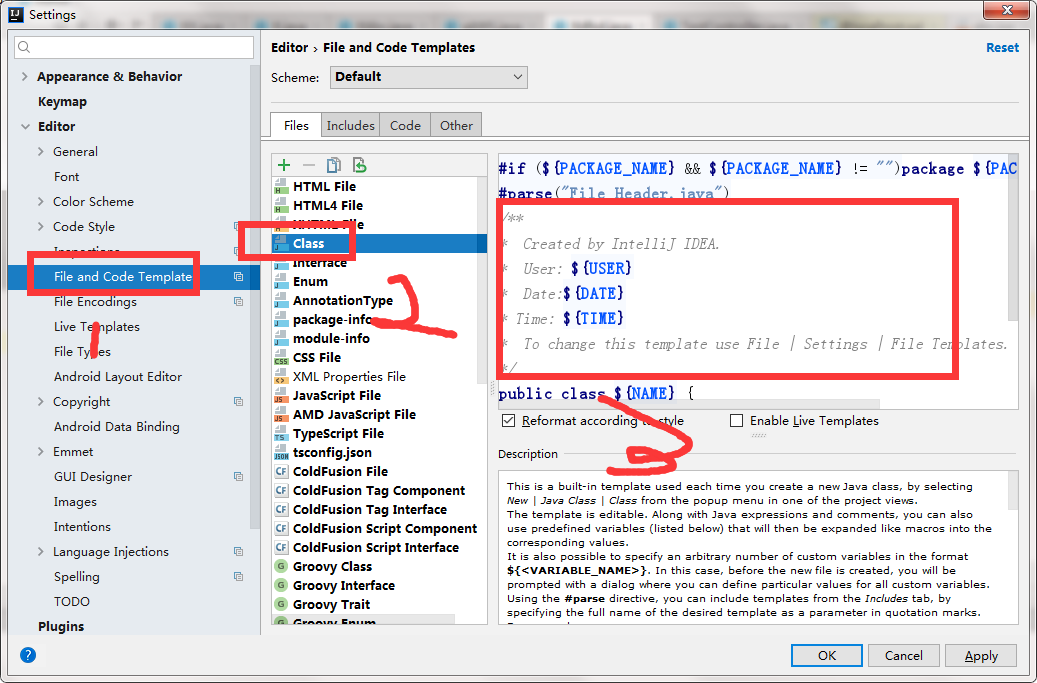
## 2.自定义设置

自定义模板配置使用

file->setting->Editor->Filr and Code Templates

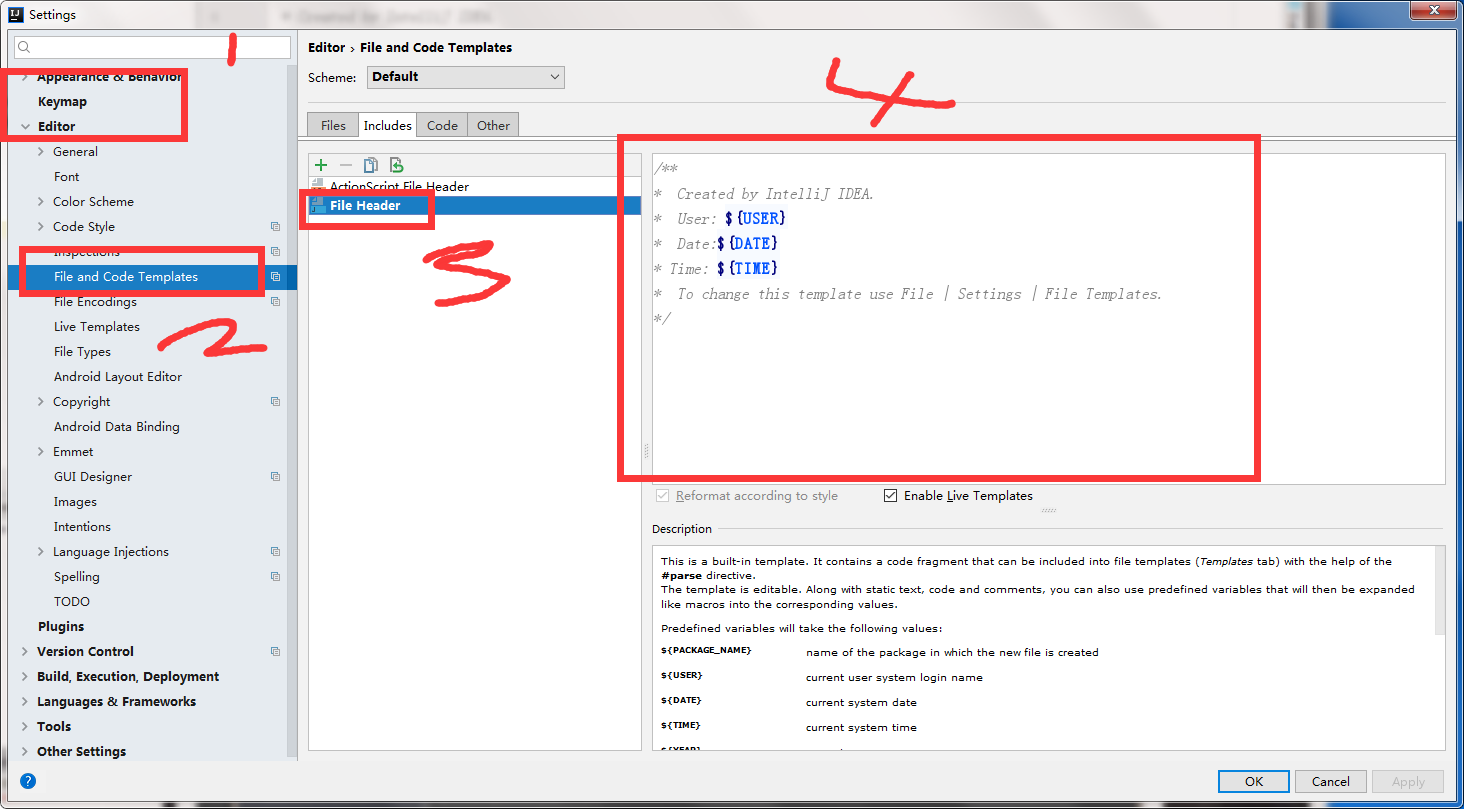
### 类注释模板

方法一：



方法二：

好多设置引入了File header文件。直接修改include的文件

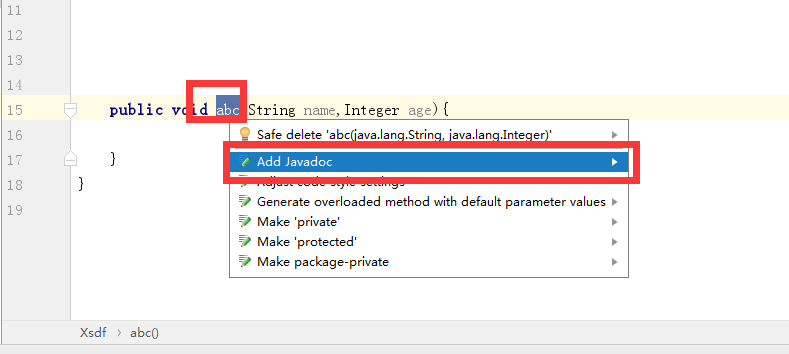


/\*\*  
\* Created by IntelliJ IDEA.  
\* User: **${USER}**\* Date:**${DATE}**\* Time: **${TIME}**\* To change this template use File | Settings | File Templates.  
\*/

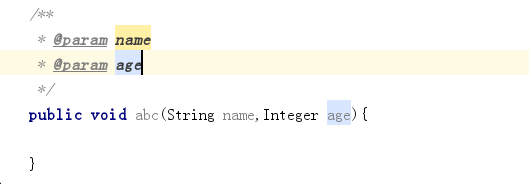
### 方法注释

简单的方式 用java doc方式

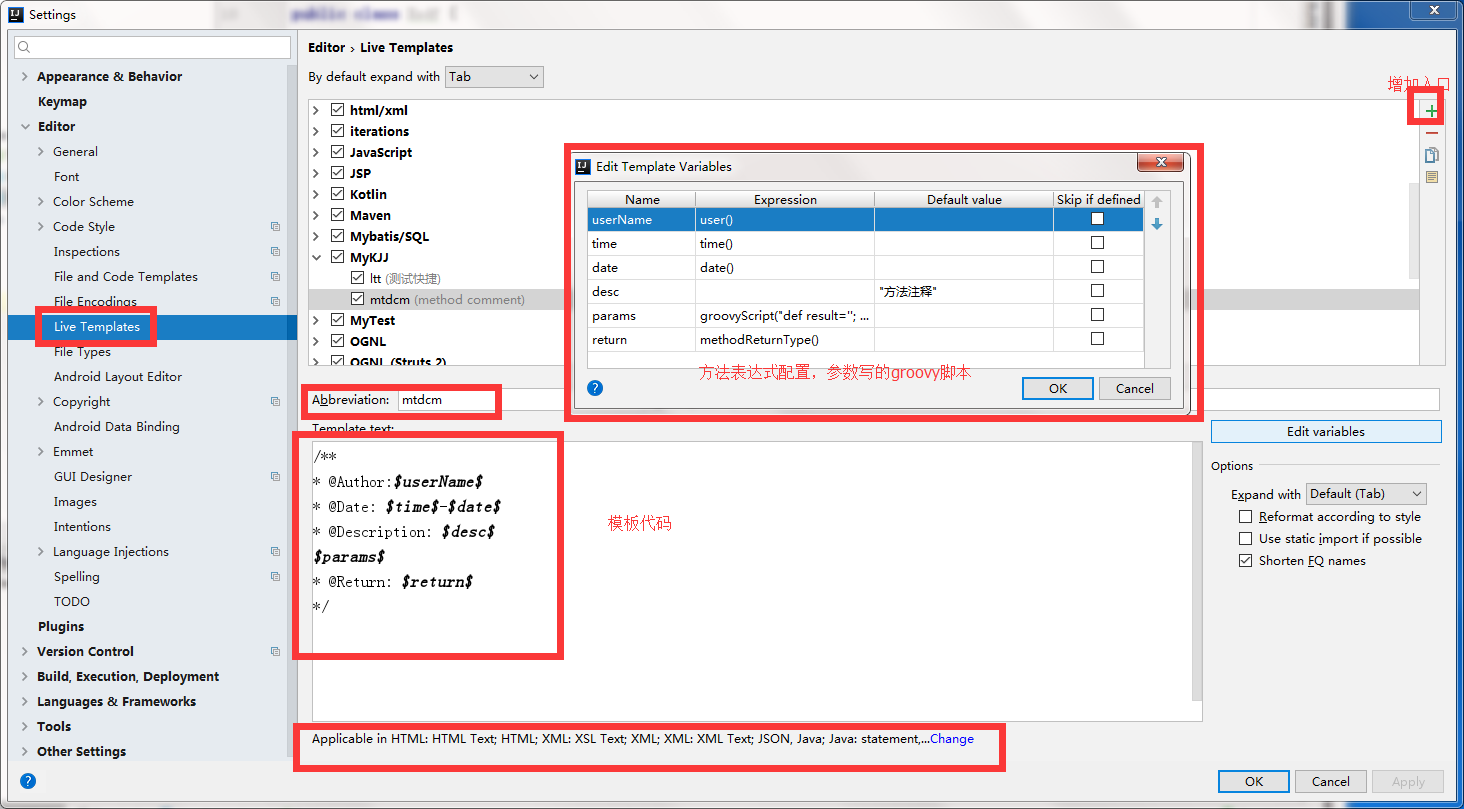
选中方法： alt +enter --》 Add javadoc



效果：



其他方式：自己可以写一个快捷键方式。



/\*\*

\* @Author:$userName$

\* @Date: $time$-$date$

\* @Description: $desc$

$params$

\* @Return: $return$

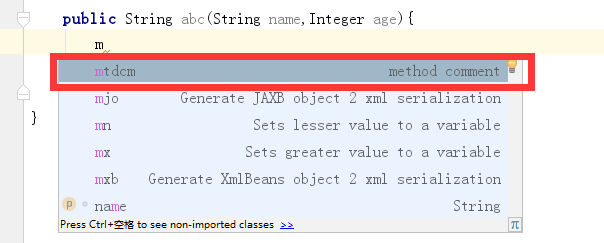
\*/

其中脚本：

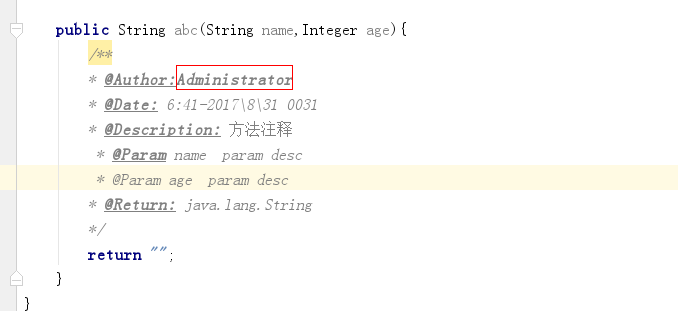
groovyScript("def result=''; def params=\"${\_1}\".replaceAll('[\\\\[|\\\\]|\\\\s]', '').split(',').toList(); for(i = 0; i < params.size(); i++) {result+=' \* @Param ' + params[i]+' param desc ' + ((i < params.size() - 1) ? '\\n\\b' : '')}; return result", methodParameters())

得在方法级内部使用，不然参数返回类型等拿不到。

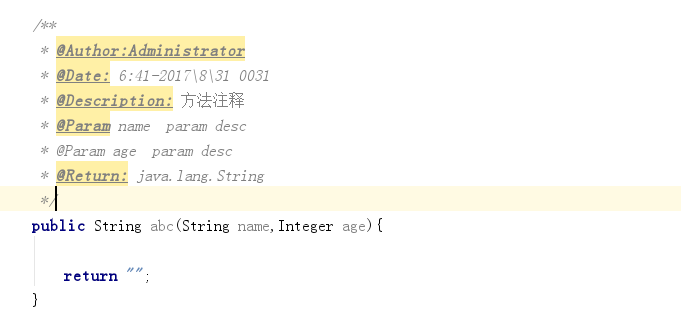
效果：



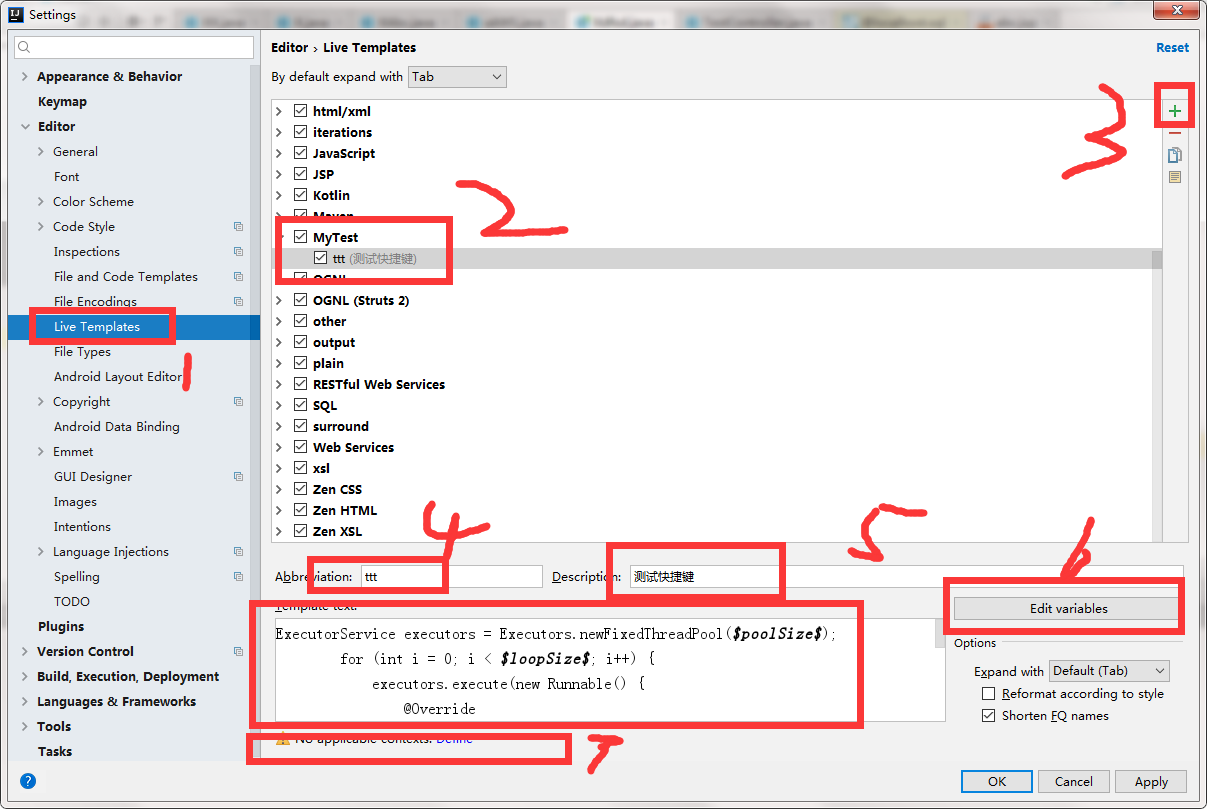
选中自定义函数 mtdcm

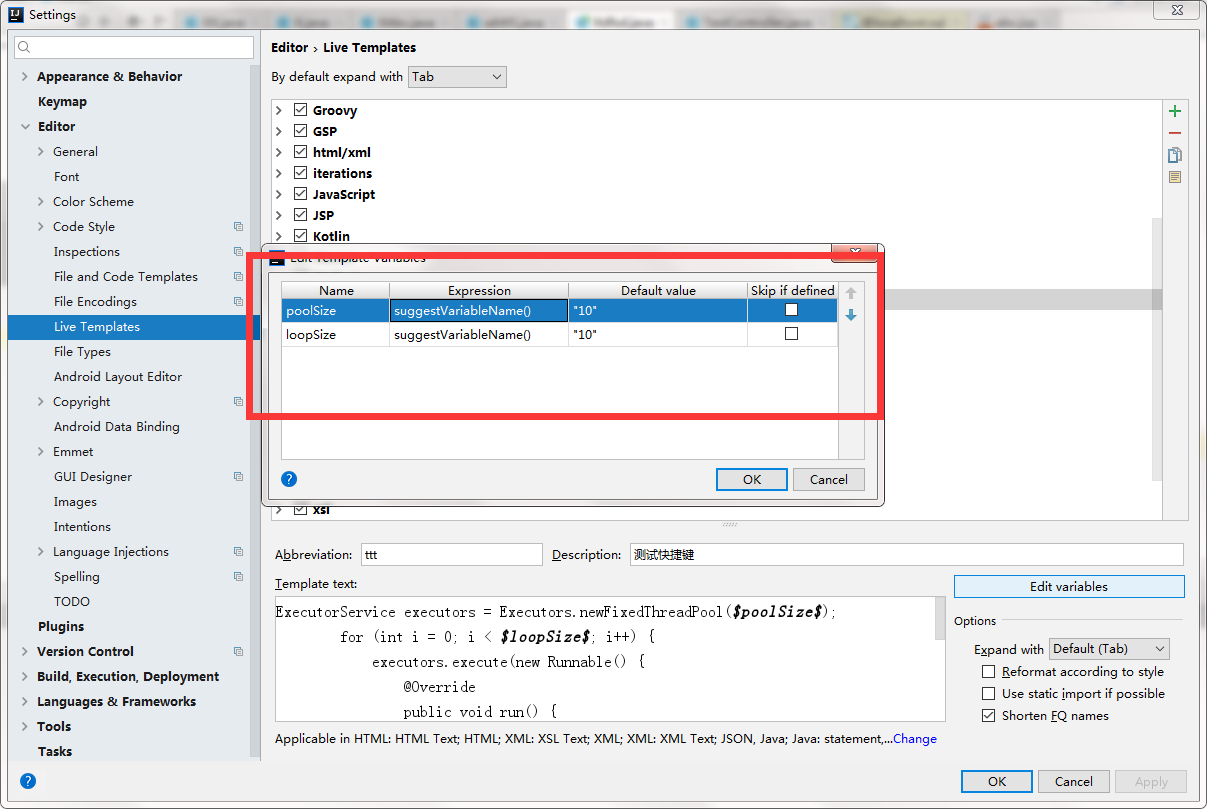


手工copy到方法上边去。



## 自定义快捷键





0.idea常用 快捷键

  **Top #10切来切去**：Ctrl+Tab

  **Top #9选你所想**：Ctrl+W

  **Top #8代码生成**：Template/Postfix +Tab

  **Top #7发号施令**：Ctrl+Shift+A

  **Top #6无处藏身**：Shift+Shift

  **Top #5自动完成**：Ctrl+Shift+Enter

  **Top #4创造万物**：Alt+Insert

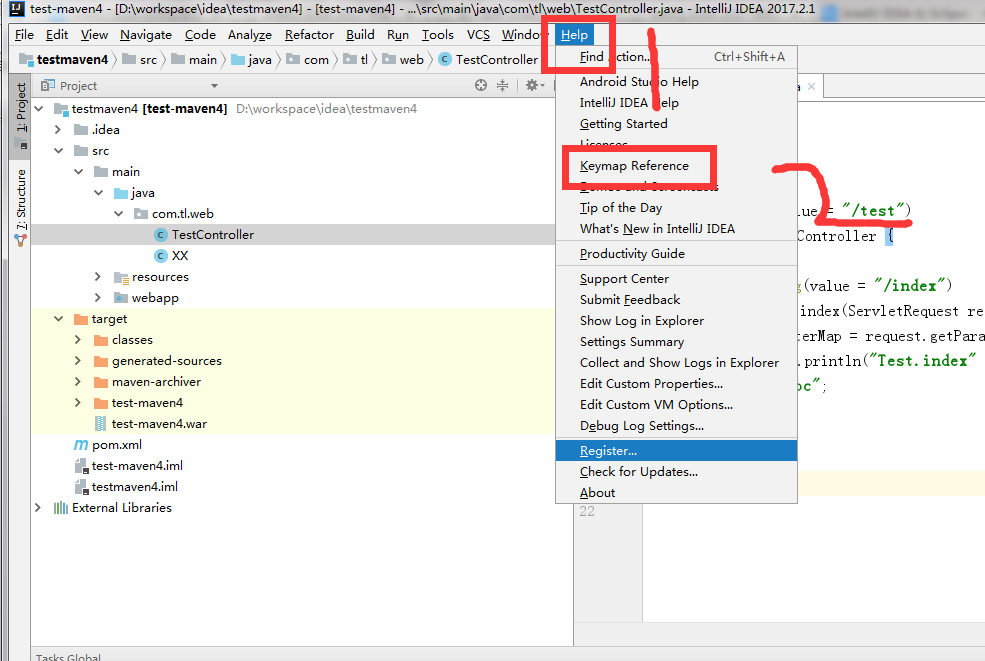
太难割舍，前三名并列吧！

  **Top #1智能补全**：Ctrl+Shift+Space

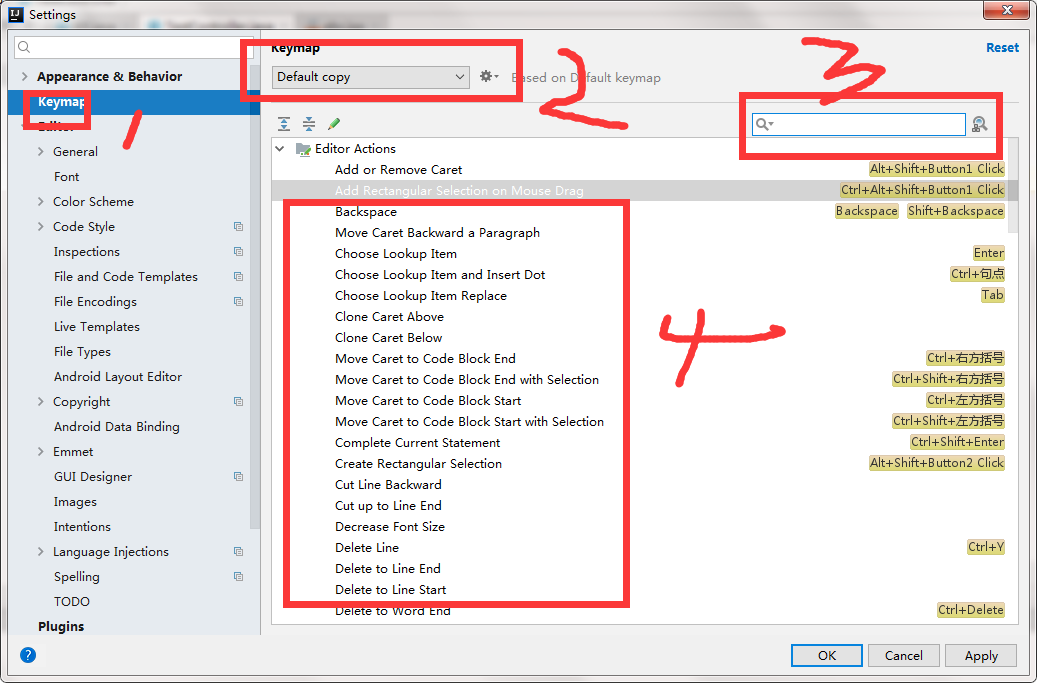
  **Top #1自我修复**：Alt+Enter

  **Top #1重构一切**：Ctrl+Shift+Alt+T

idea 快捷键介绍

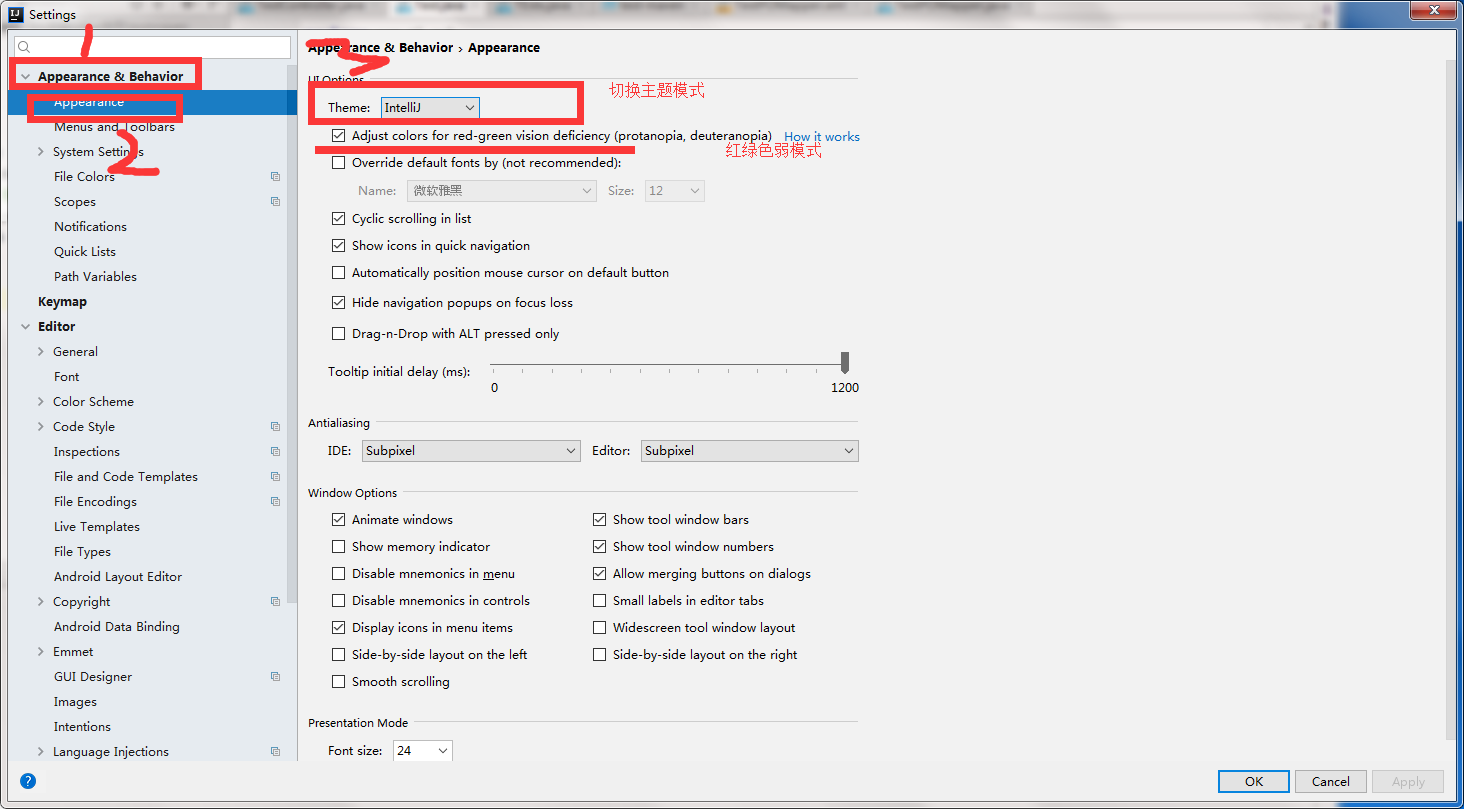


快捷键修改

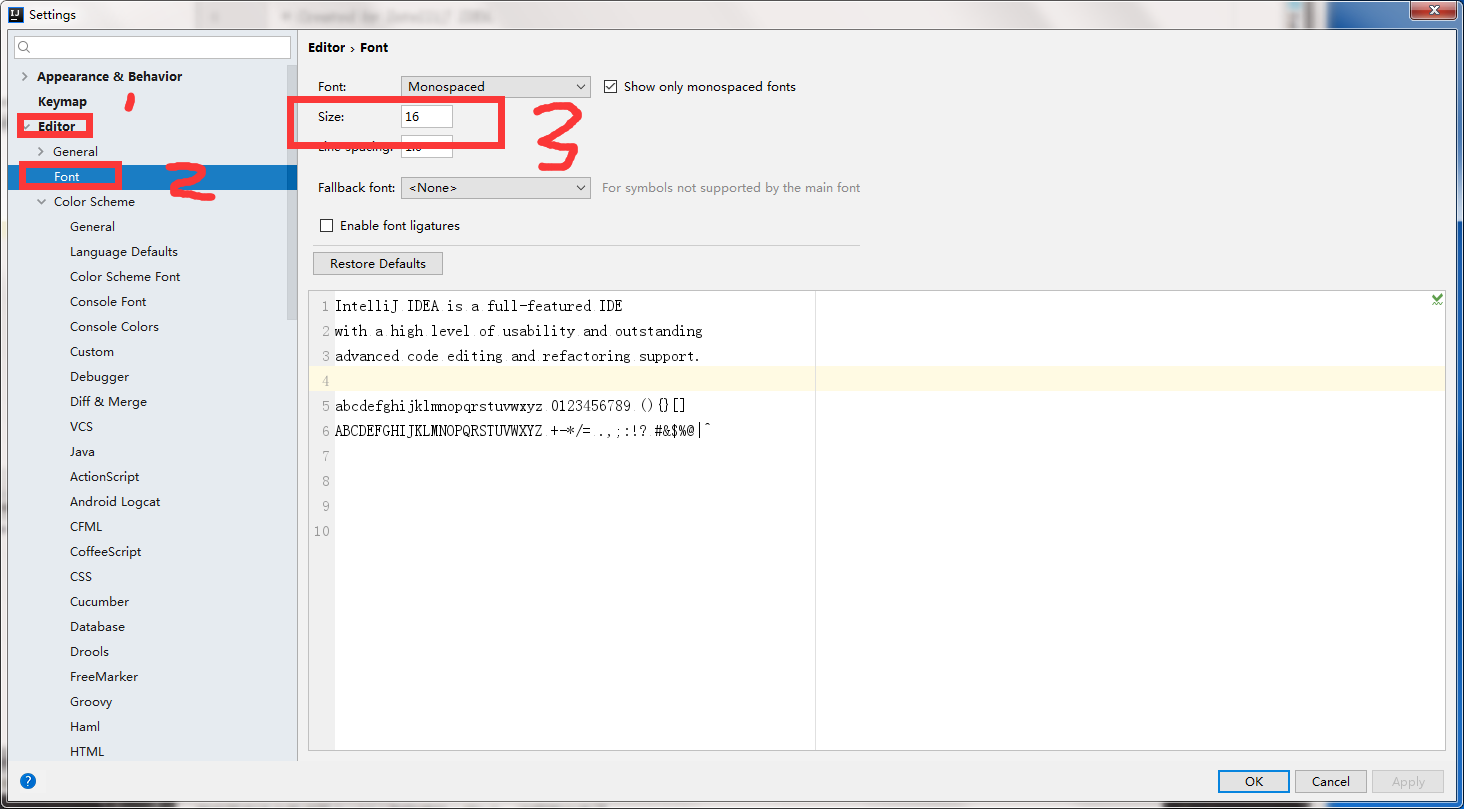


其他常用设置：

修改主题



调整字体大小：



插件名称 插件介绍 官网地址   
Key promoter 快捷键提示 [https://plugins.jetbrains.com/plugin/4455?pr=idea](https://plugins.jetbrains.com/plugin/4455?pr=idea" \t "_blank)

CamelCase 驼峰式命名和下划线命名交替变化 [https://plugins.jetbrains.com/plugin/7160?pr=idea](https://plugins.jetbrains.com/plugin/7160?pr=idea" \t "_blank)

CheckStyle-IDEA 代码规范检查 [https://plugins.jetbrains.com/plugin/1065?pr=idea](https://plugins.jetbrains.com/plugin/1065?pr=idea" \t "_blank)

FindBugs-IDEA 潜在 Bug 检查 [https://plugins.jetbrains.com/plugin/3847?pr=idea](https://plugins.jetbrains.com/plugin/3847?pr=idea" \t "_blank)

MetricsReloaded 代码复杂度检查 [https://plugins.jetbrains.com/plugin/93?pr=idea](https://plugins.jetbrains.com/plugin/93?pr=idea" \t "_blank)

Statistic 代码统计 [https://plugins.jetbrains.com/plugin/4509?pr=idea](https://plugins.jetbrains.com/plugin/4509?pr=idea" \t "_blank)

JRebel Plugin 热部署 [https://plugins.jetbrains.com/plugin/?id=4441](https://plugins.jetbrains.com/plugin/?id=4441" \t "_blank)

CodeGlance 在编辑代码最右侧，显示一块代码小地图 [https://plugins.jetbrains.com/plugin/7275?pr=idea](https://plugins.jetbrains.com/plugin/7275?pr=idea" \t "_blank)

GsonFormat 把 JSON 字符串直接实例化成类 [https://plugins.jetbrains.com/plugin/7654?pr=idea](https://plugins.jetbrains.com/plugin/7654?pr=idea" \t "_blank)

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。 [https://plugins.jetbrains.com/plugin/6546?pr=idea](https://plugins.jetbrains.com/plugin/6546?pr=idea" \t "_blank)