Intellj Idea 使用介绍

课程计划：

讲解git的使用

IDEA的安装

集成JDK（1.8）

常用的样式及配置介绍

idea的web工程的建立

运行web项目(使用tomcat)

idea的maven工程建立（jar pom war）

SVN的集成

GIT的集成

debug的调试

导入eclipse项目集成到idea中

常用快捷键的说明

动态自定义模板(sysout)

# idea的安装

## IDEA的介绍

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公司的产品，这家公司总部位于捷克共和国的首都布拉格，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。免费版只支持Java等少数语言。

IDEA 分为商业版和免费版本。

商业版：功能强大。

免费版：功能单一。

以下的图 展示免费版没有而商业版却有的功能。



### 功能特色

IDEA所提倡的是智能编码，是减少程序员的工作。

* 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。

* 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视图中，你还可以选择多种的视图方式。

* 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，在版本恢复时你可以很容易的将其恢复。

* JUnit的完美支持
* 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。

* 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。

* 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，因为它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。

* XML的完美支持

xml全提示支持：所有流行框架的xml文件都支持全提示，谁用谁知道。

* **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。

* 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。

* 对JSP的完全支持

不需要任何的插件，完全支持JSP。

* 智能编辑

代码输入过程中，自动补充方法或类。

* EJB支持

不需要任何插件完全支持EJB(6.0 支持EJB3.0)

* 列编辑模式

用过UtralEdit的肯定对其的列编辑模式赞赏不已，因为它减少了很多无聊的重复工作，而IDEA完全支持该模式，从而更加提高了编码效率。

* 预置模板

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.psvm

* 完美的自动代码完成

智能检查类中的方法，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。

* 版本控制完美支持

集成了目前市面上常见的所有版本控制工具插件，包括clearcase ,CVS,git、svn、github，让开发人员在编程的工程中直接在intellij idea里就能完成代码的提交、检出、解决冲突、查看版本控制服务器内容等等。

* 不使用代码的检查

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。

* 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";

* 正则表达式的查找和替换功能

查找和替换支持正则表达式，从而提高效率。

* 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。

* 程序员意图支持

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。

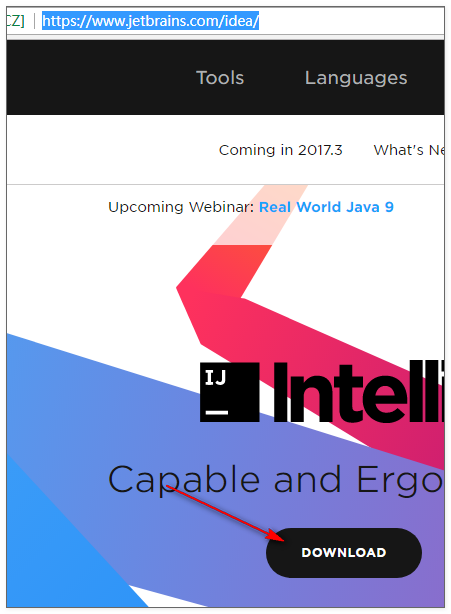
不需要再使用CTR+S 不停的保存代码

## IDEA的安装

### 下载

下载地址：https://www.jetbrains.com/idea/

idea kotlin



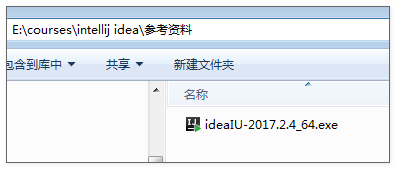
2.5版本

选择商业版的：如图



### 安装

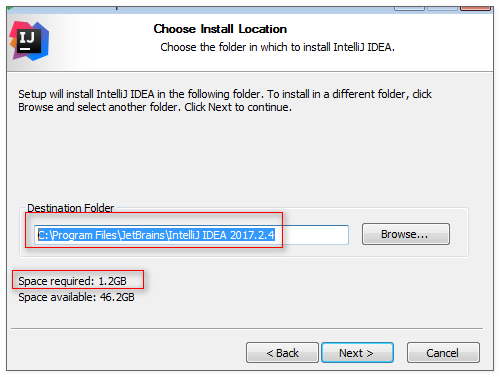
#### 第一步：执行exe文件



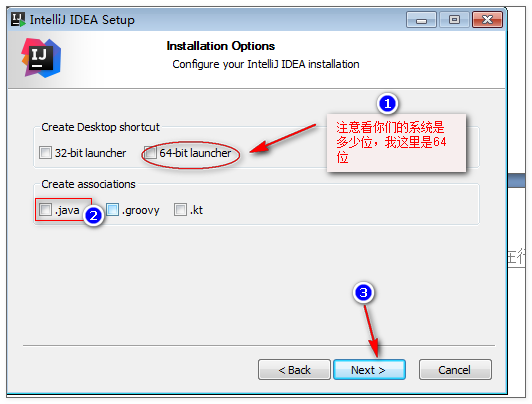
#### 第二步：下一步



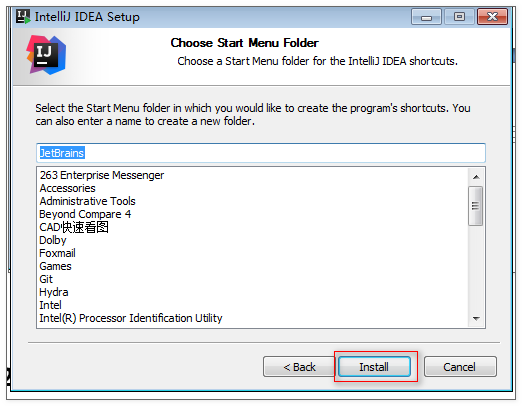
#### 第三步：选择安装的目录

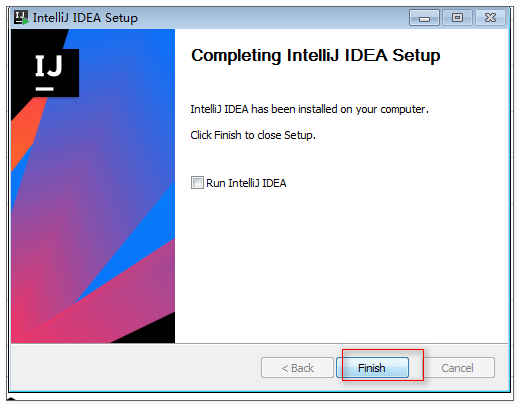


#### 第四步：选择安装系统位数和关联的文件后缀

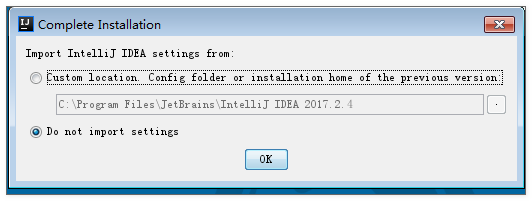


#### 第五步：安装



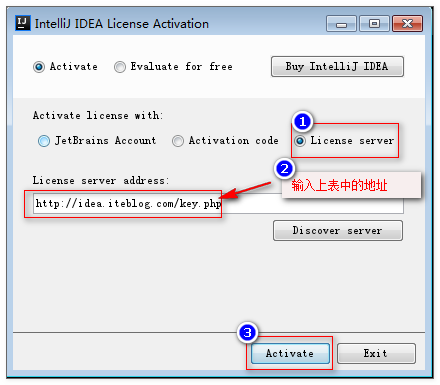


### 运行IDEA及设置授权

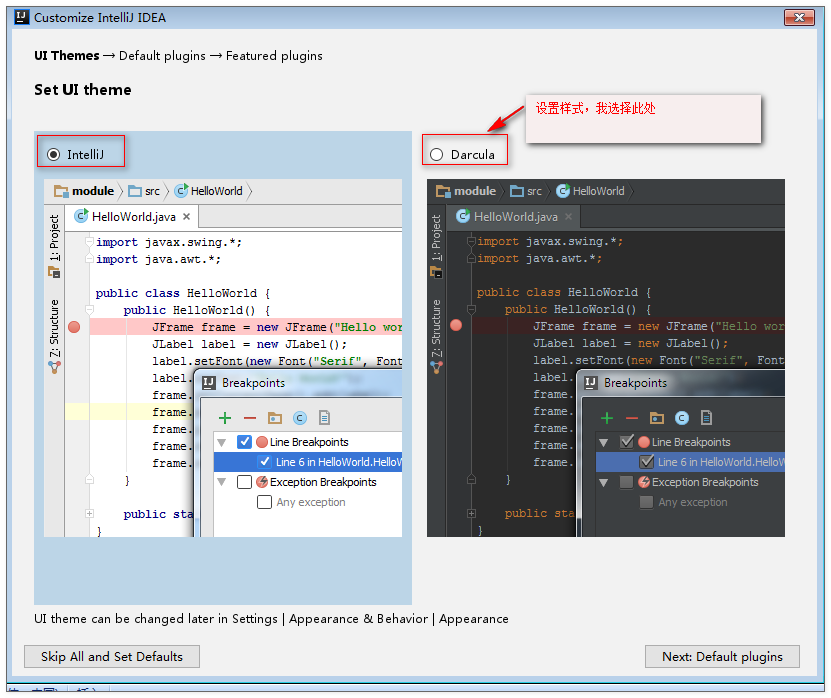


填入以下其中任何一种：

|  |
| --- |
| 填入下面的license server:  　　http://intellij.mandroid.cn/ 　　http://idea.imsxm.com/ 　　http://idea.iteblog.com/key.php |



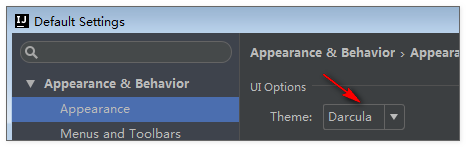
### 设置样式



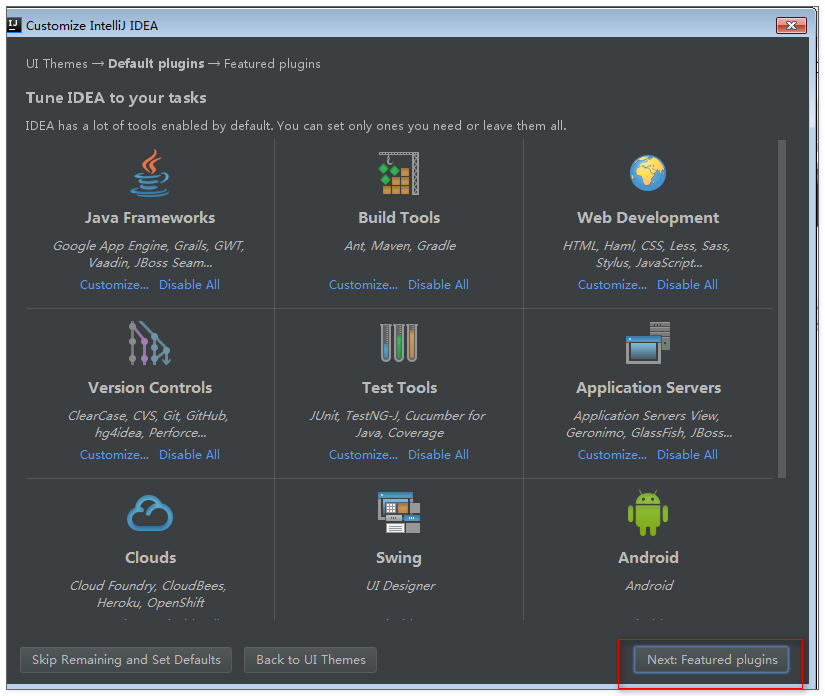
当然：样式的设置可以如下图所示的路径中设置



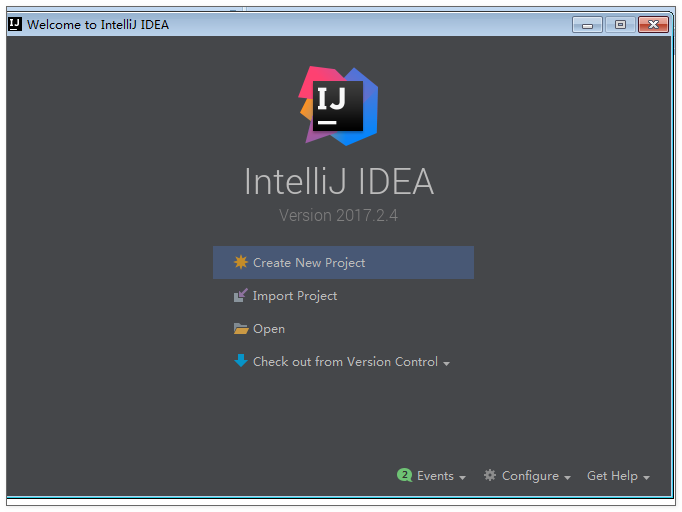
如图：这里面设置样式。settings--->apprearance&Behavior



一直点next即可。

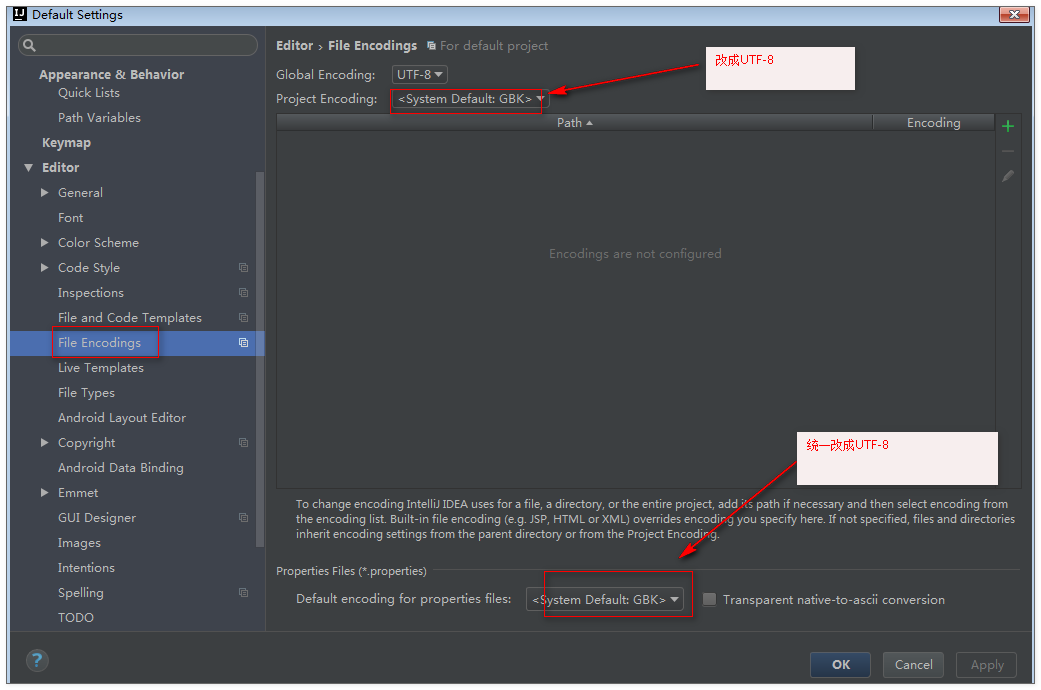


### 运行成功，创建工程



### 设置字符编码

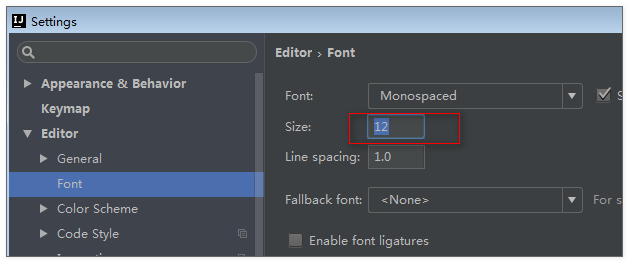
在configure-->setting 中设置所有的文件的字符编码



### 设置字体大小

File-->settings-->editor--->font

设置字体的大小。



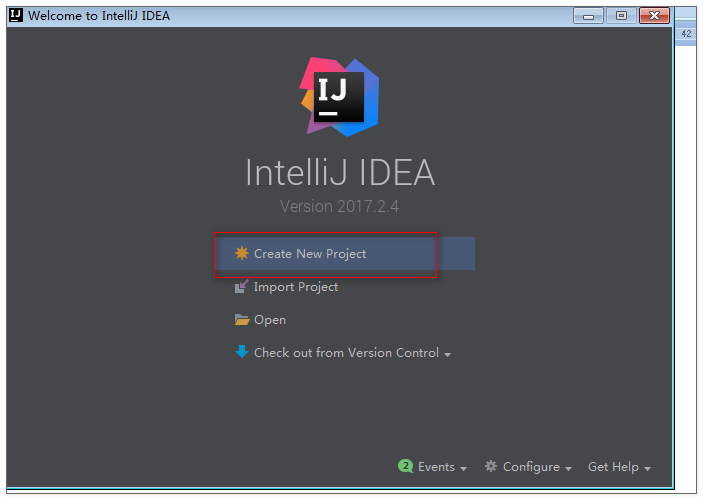
# idea集成JDK

## 创建工程

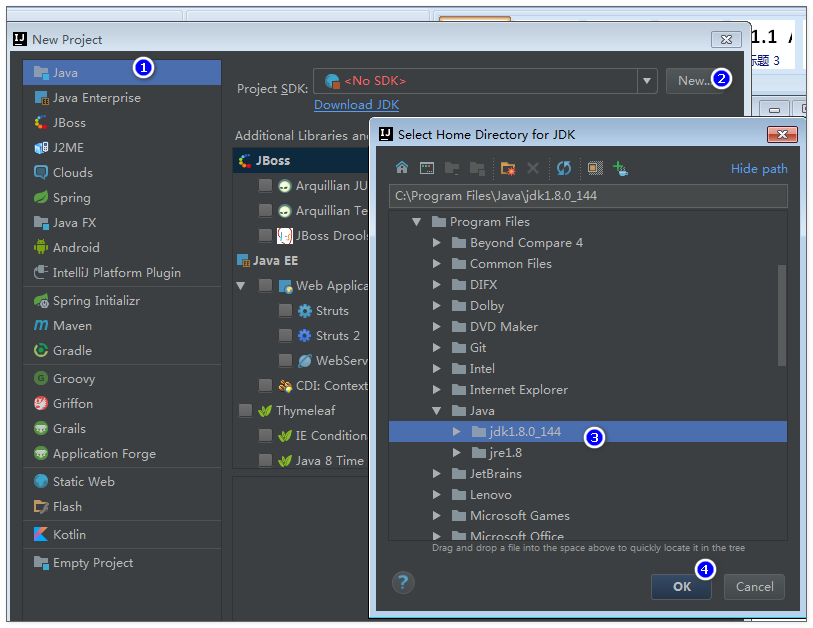
如图所示：

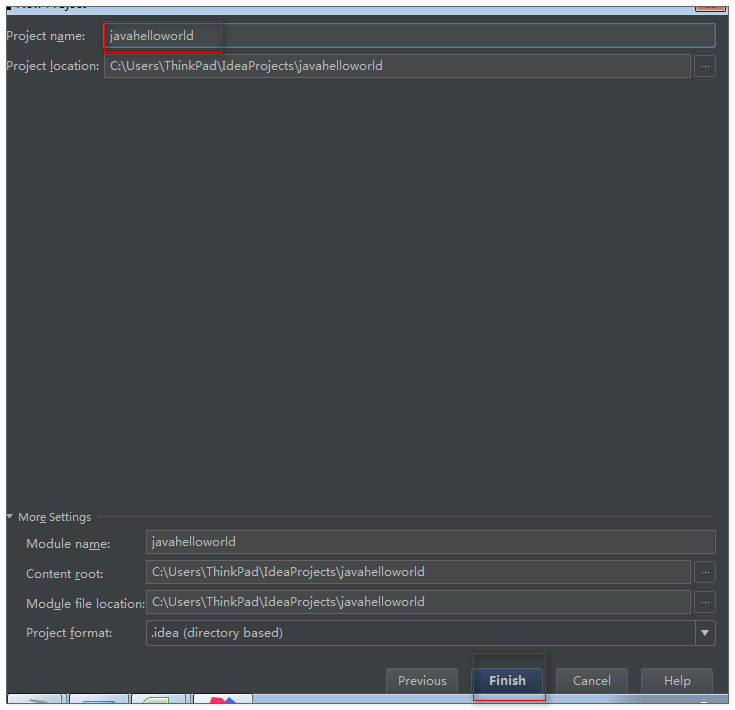
在idea中并没有类似于eclipse中的workspace的概念。

只有工程的概念。但是idea有模块的概念。



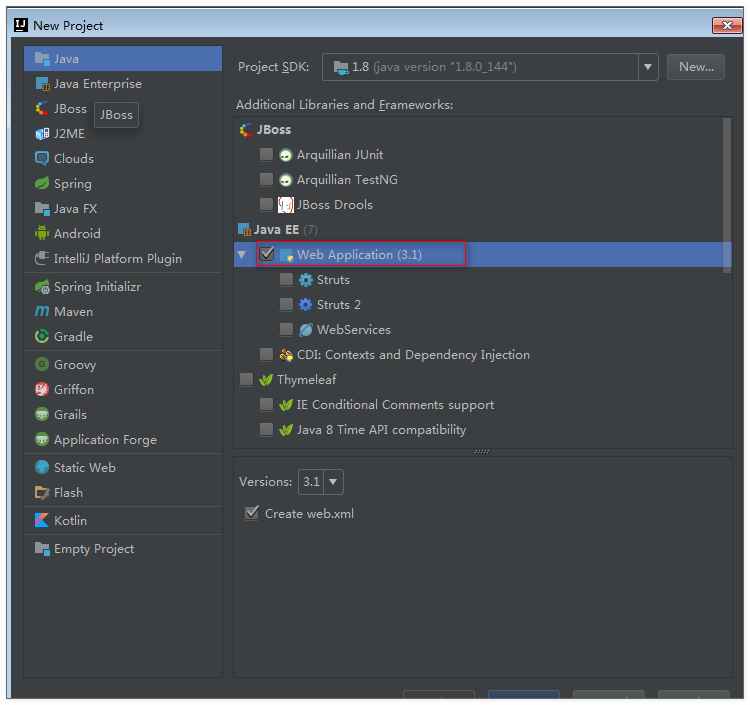
## 设置JDK的配置





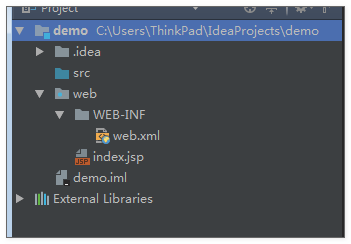
# 创建web工程

## 创建web工程



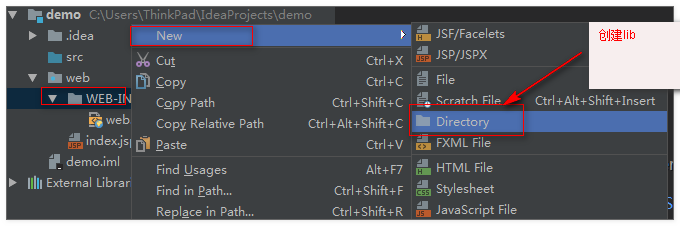
勾选web application表示创建的web工程

创建成功后：目录显示如下：

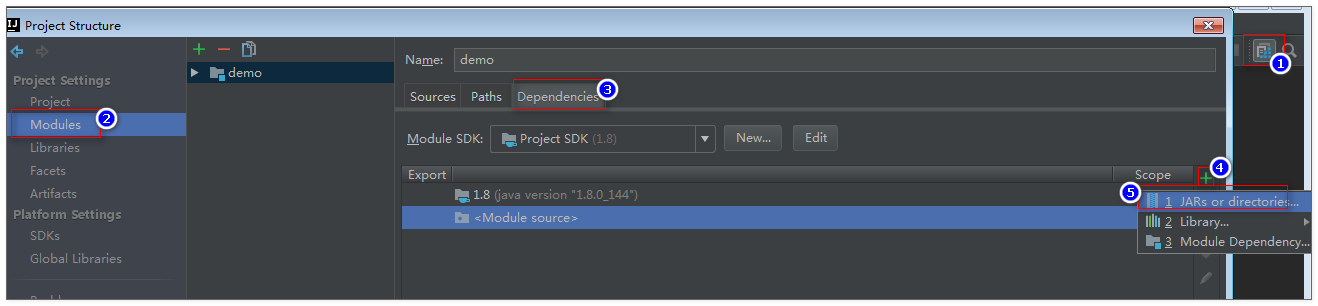


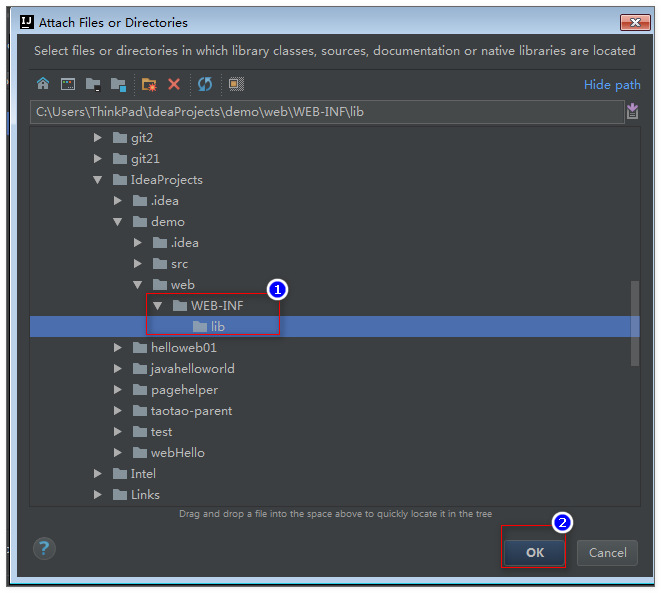
## 添加jar包

### 创建目录

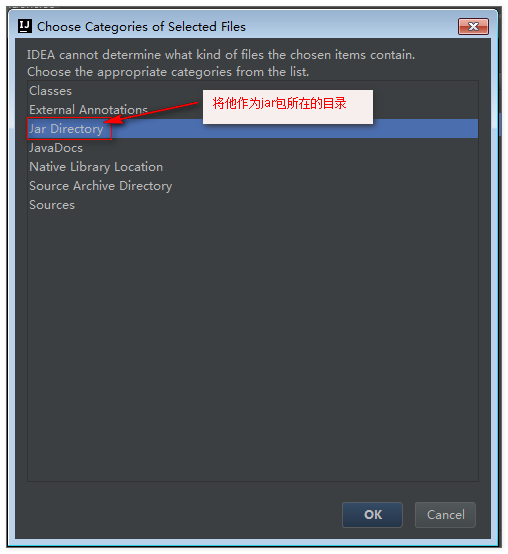


### 添加jar包到工程中



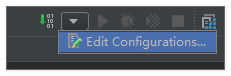


这样，如图，将jar包放入其中便可以使用了。类似于：build path

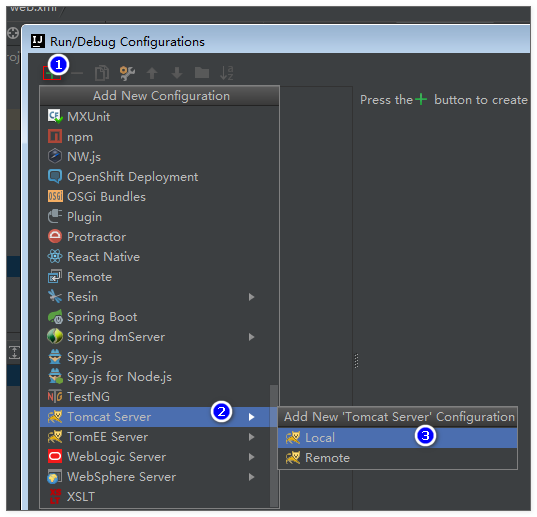


## 启动tomcat发布运行web工程

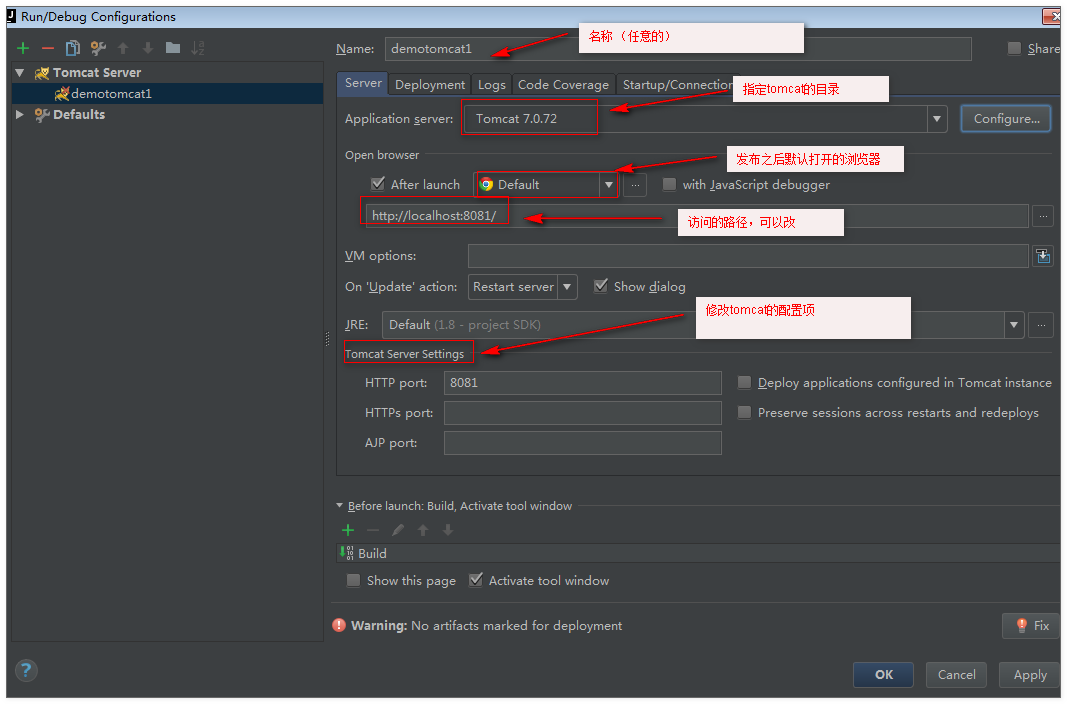
* 点击编辑

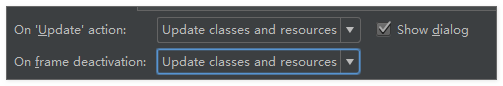


* 添加tomcat server



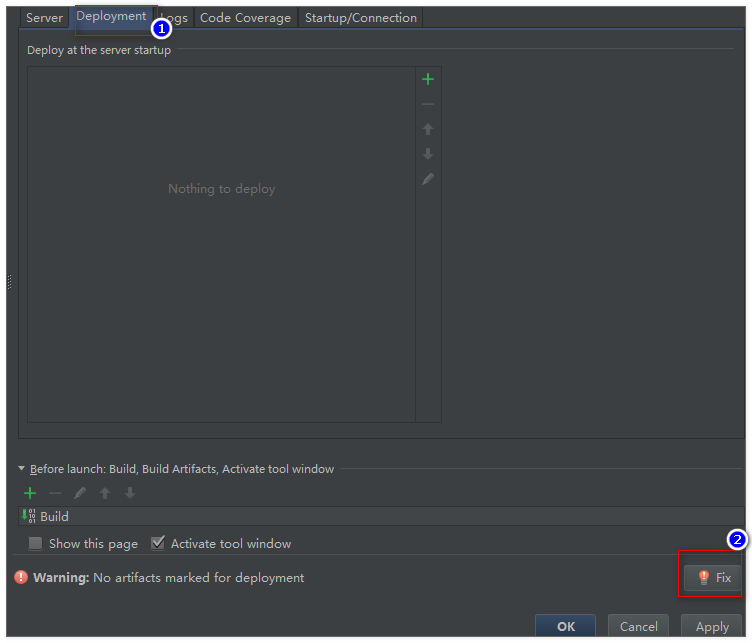
* 添加tomcat server



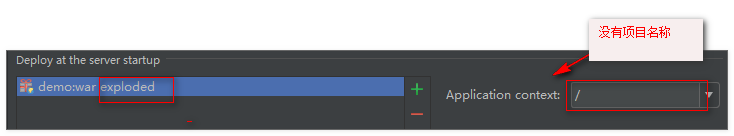


这里的配置可以用于:当jsp页面修改了之后,重新编译class

* 添加部署的war包



* 设置访问的路径



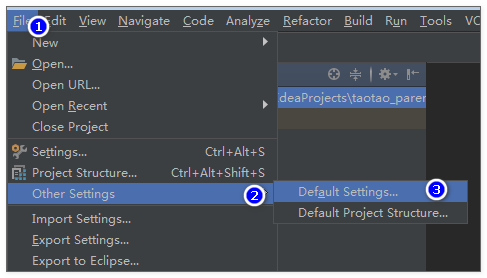
* 点击运行：

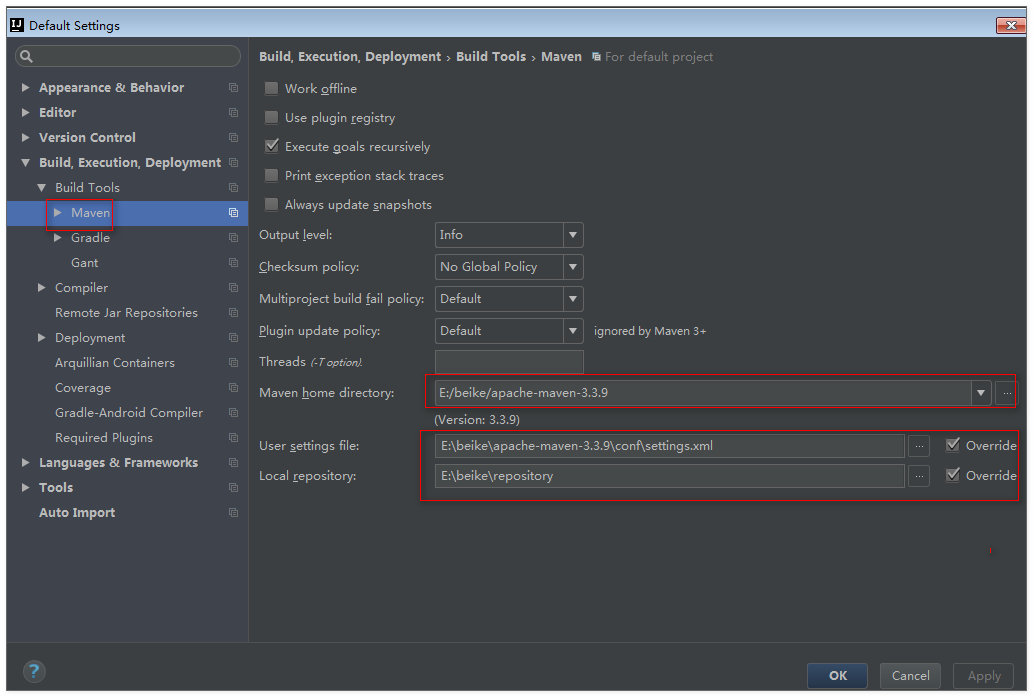


启动成功并测试成功。

# 创建maven工程

## 设置maven的环境





如图 ：

设置maven的主目录。

设置配置文件

设置本地仓库

## 设置源为maven的版本1.8

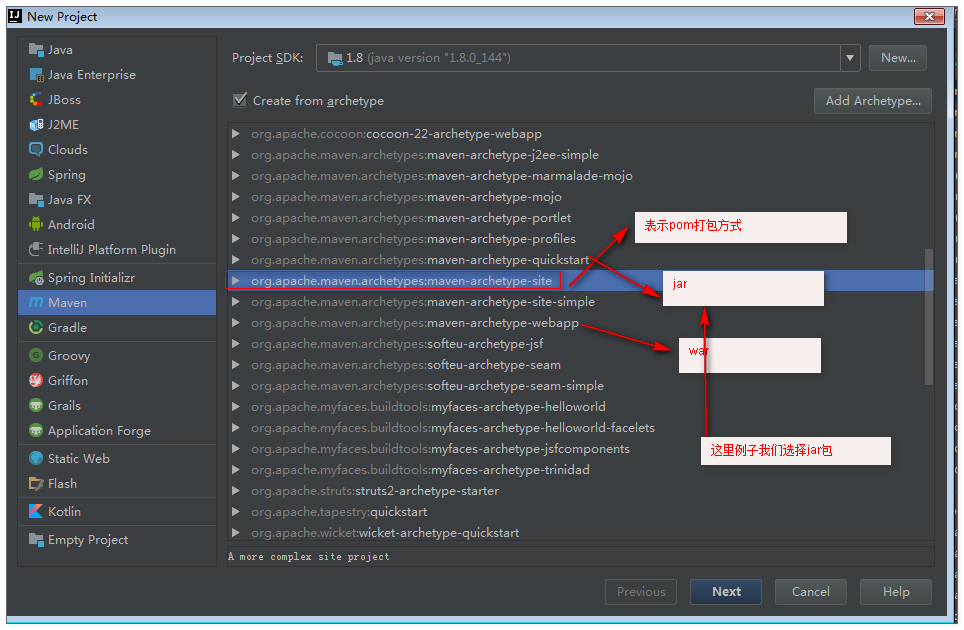
默认情况下idea设置为JDK源为1.5 需要在maven中配置properties

|  |
| --- |
| <properties>  <maven.compiler.source>1.8</maven.compiler.source>  <maven.compiler.target>1.8</maven.compiler.target>  </properties> |

## 创建打包方式为jar的maven工程

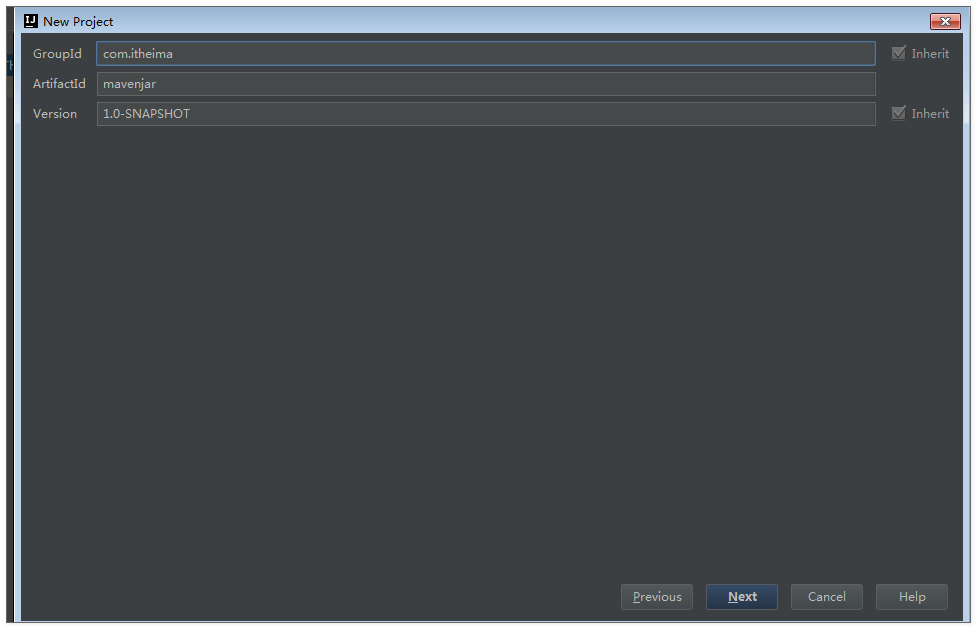
### 创建maven项目选择quickstart

选择maven项目，并选择骨架

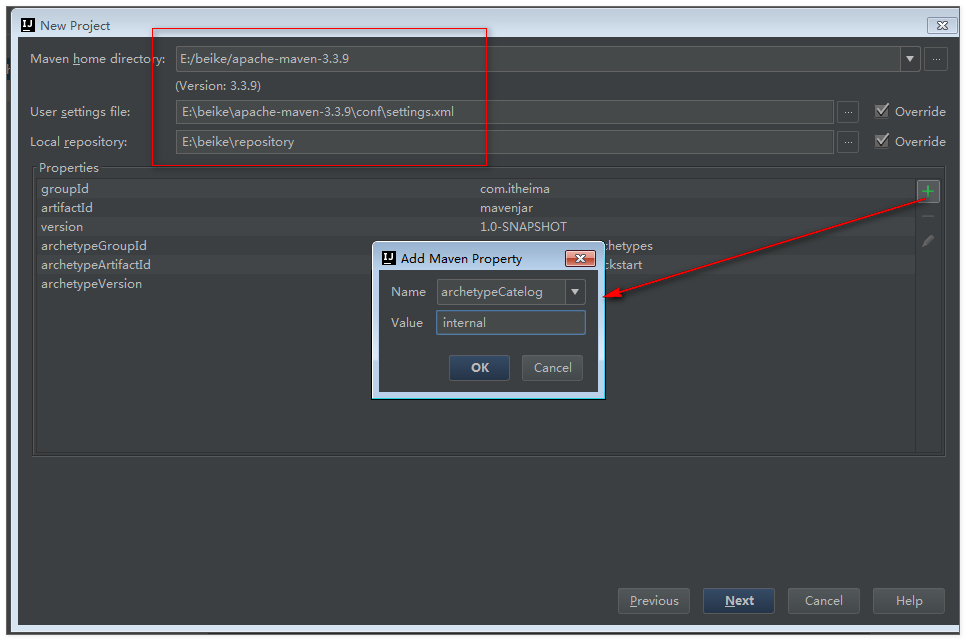


此处选择jar包

### 填写组织和项目名



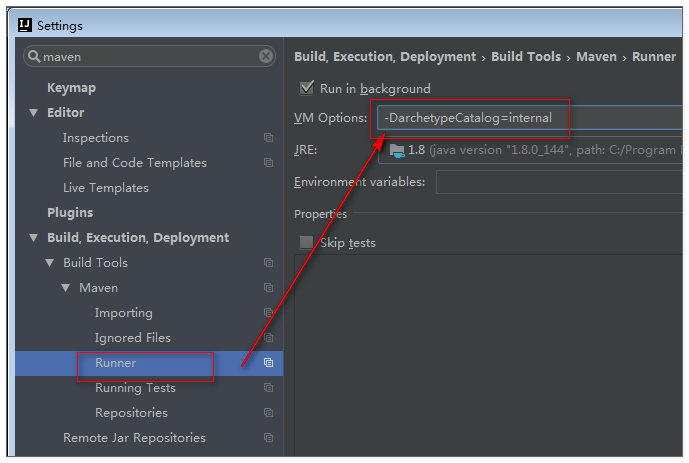
### 设置archetypeCatelog=internal



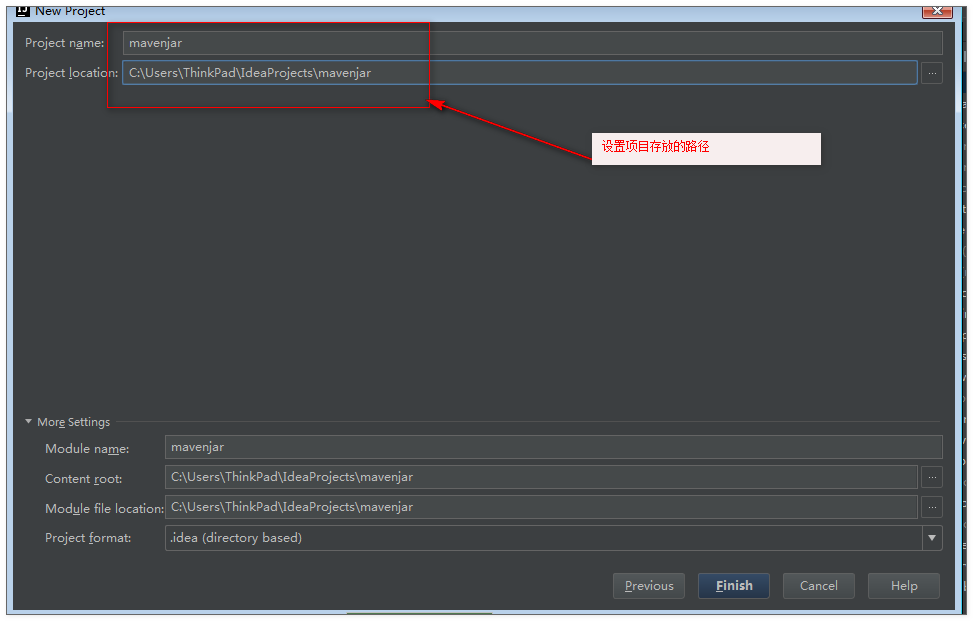
此处的设置的作用就是：让maven项目更加快速的创建项目。如果不设置每次都需要联网下载元数据。设置此配置之后，第一次下载好了之后直接从本地仓库取。

以上，每次创建工程都需要设置，比较麻烦，可以统一设置，将来创建工程不需要添加配置了。配置如下：在settings中。

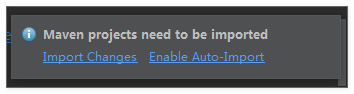
-DarchetypeCatalog=local



### 选择创建的目录



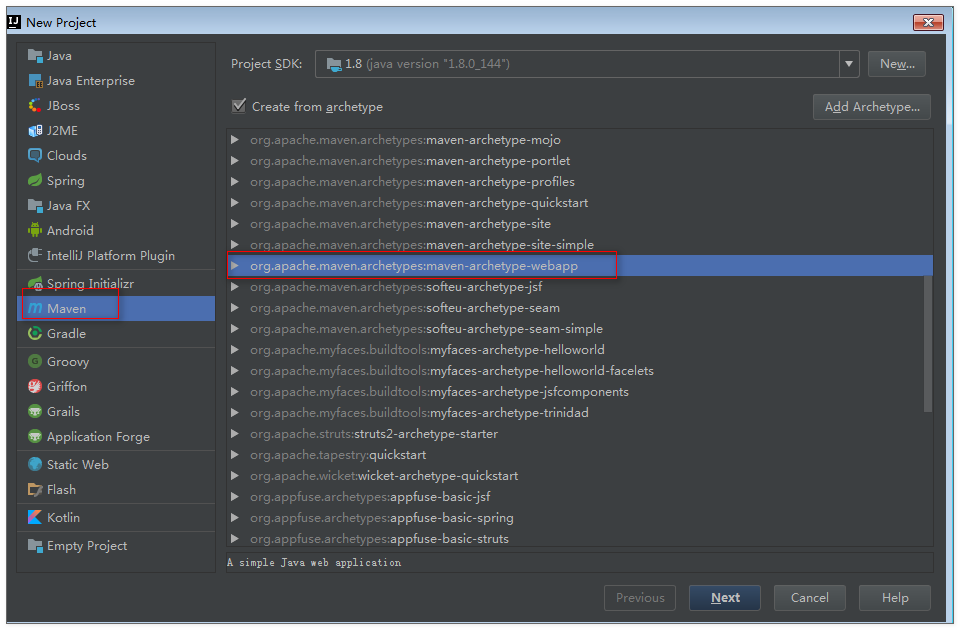
### 开启自动导入



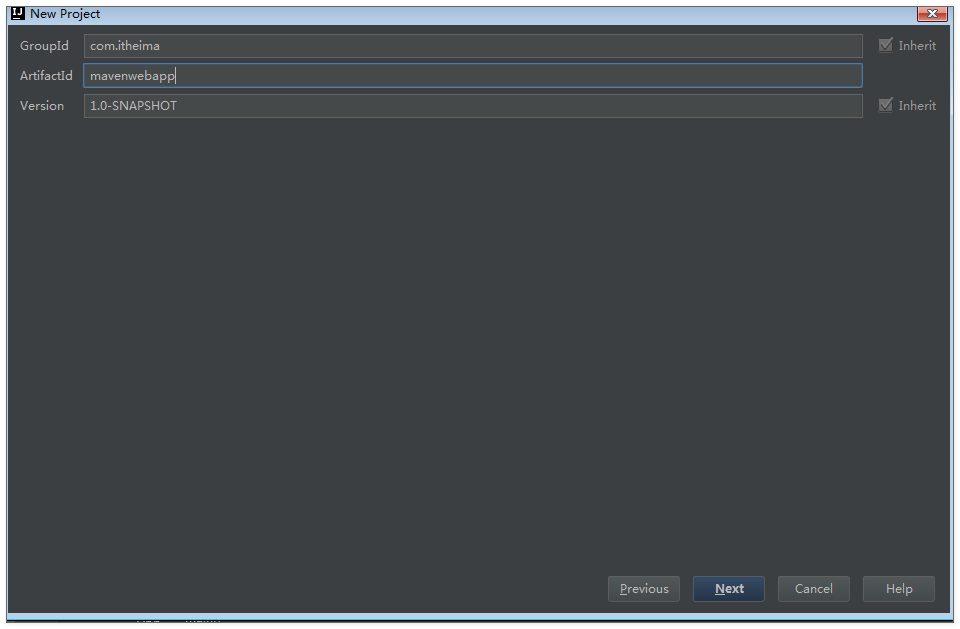
至此maven工程创建成功。

## 创建打包方式为war包的maven项目

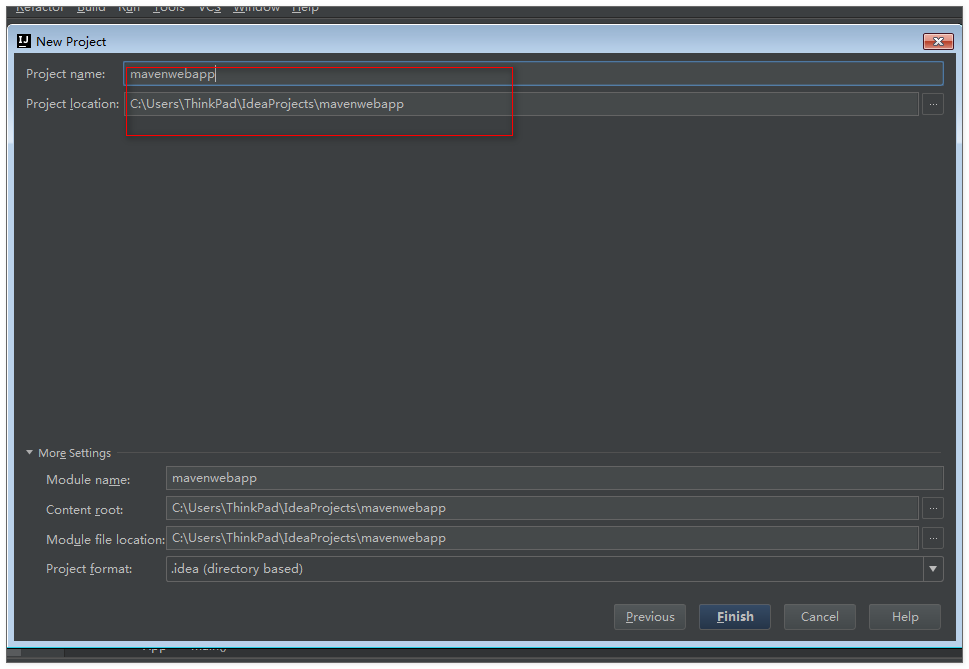
### 创建maven项目选择webapp



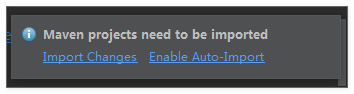
### 填写组织和项目名



### 设置文件所在路径



### 开启自动导入



创建成功

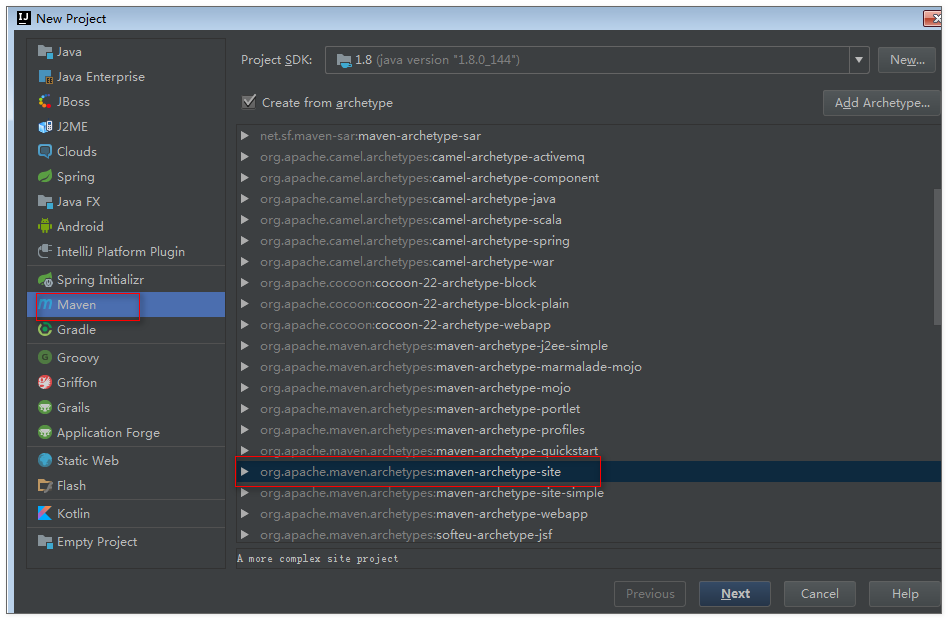


## 创建打包方式为pom的maven项目

### 创建maven项目选择site

#### 选择完整site

选择如图这种，会生成完整的pom的站点目录结构。



pom.xml:

|  |
| --- |
| <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.itheima</groupId>  <artifactId>mavensitemore</artifactId>  <version>1.0-SNAPSHOT</version>  <url>http://maven.apache.org/</url>  <inceptionYear>2001</inceptionYear>    <distributionManagement>  <site>  <id>website</id>  <url>scp://webhost.company.com/www/website</url>  </site>  </distributionManagement>   <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>   <build>  <plugins>  <plugin>  <artifactId>maven-site-plugin</artifactId>  <configuration>  <locales>en,fr</locales>  </configuration>  </plugin>  </plugins>  </build> </project> |

#### 选择简单simple-site

pom.xml:

|  |
| --- |
| <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>   <groupId>com.itheima</groupId>  <artifactId>mavensite</artifactId>  <version>1.0-SNAPSHOT</version>  <packaging>pom</packaging>   <distributionManagement>  <site>  <id>website</id>  <url>scp://webhost.company.com/www/website</url>  </site>  </distributionManagement>   <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties> </project> |

按需求选择，这里推荐使用simple-site 。

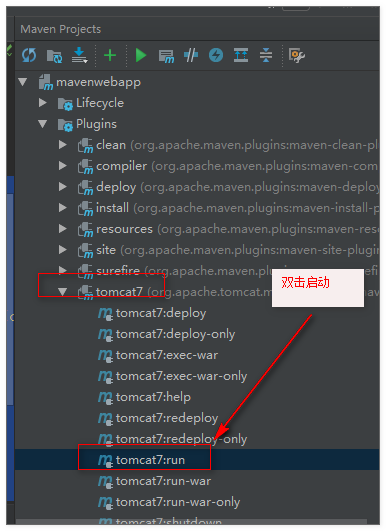
# 使用tomcat插件启动maven

## 配置tomcat插件

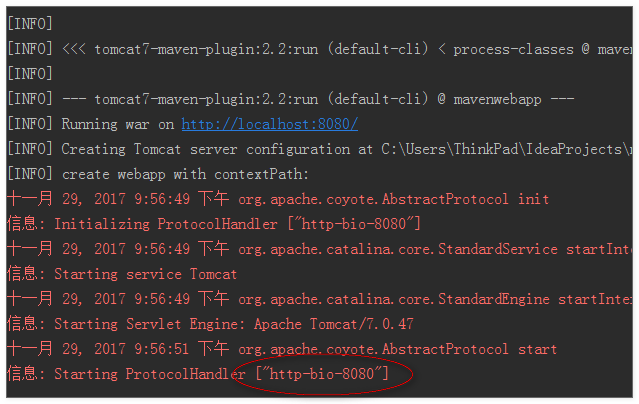
在pom.xml中配置build:

|  |
| --- |
| <build>  <finalName>mavenwebapp</finalName>  <plugins>  <plugin>  <groupId>org.apache.tomcat.maven</groupId>  <artifactId>tomcat7-maven-plugin</artifactId>  <configuration>  <port>8080</port>  <path>/</path>  </configuration>  </plugin>   </plugins> </build> |

## 启动项目

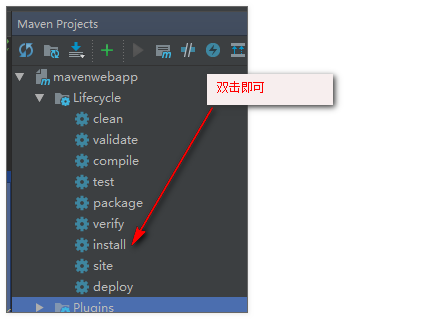


## 测试即可

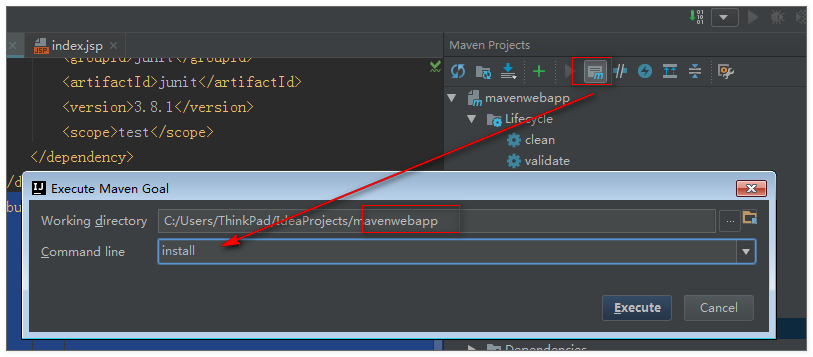


# 使用maven命令

执行安装：



或者：



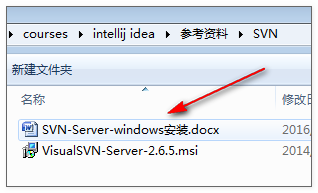
其他命令类似，这里不再讨论。

# SVN集成使用

## SVN基础环境准备

### 安装SVN服务器

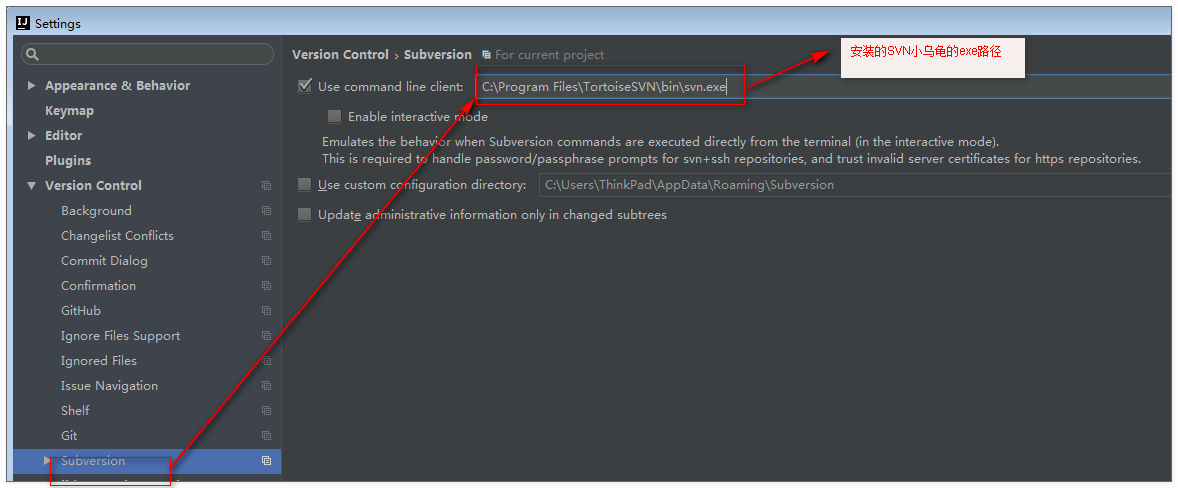
参考资料：



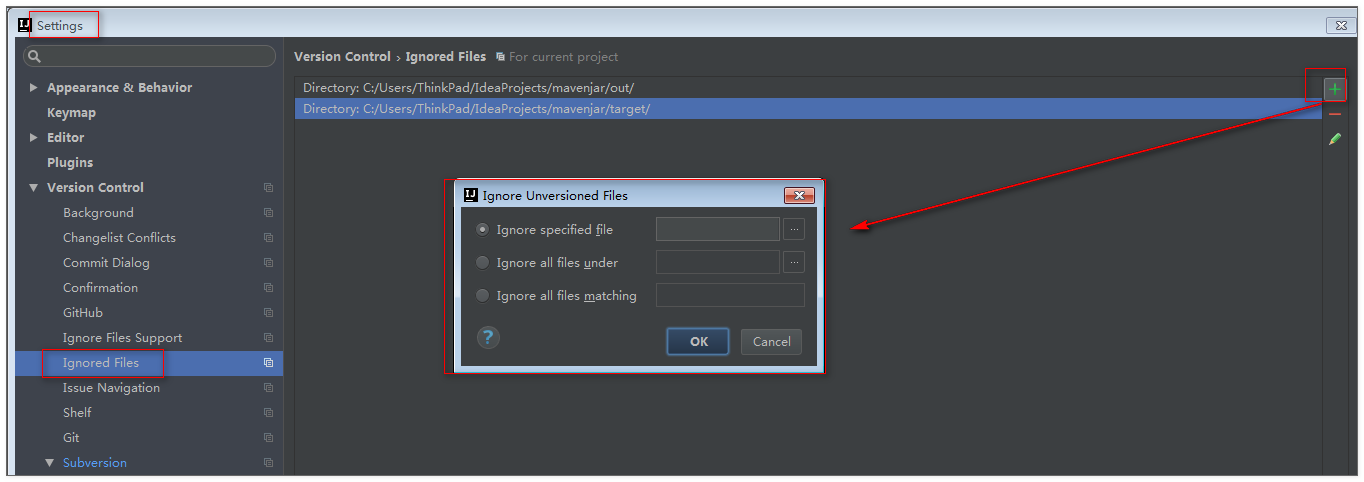
### 安装SVN小乌龟

略过

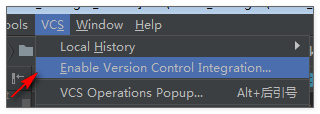
## 开启SVN集成环境



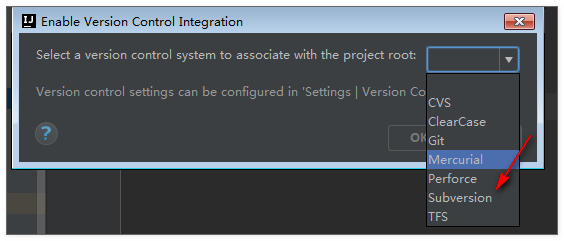
## 添加忽略文件



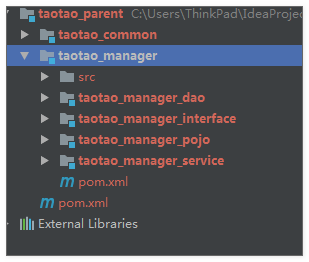
## 开启版本控制



选择使用：SVN

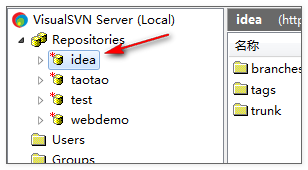


开启之后:颜色变红了



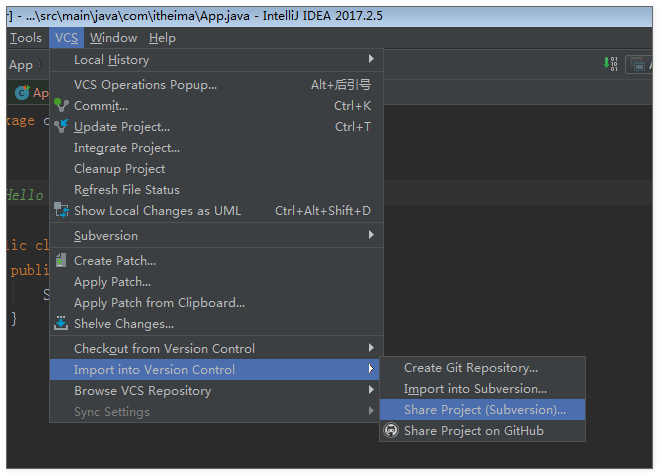
## 提交代码到SVN服务器上

### 创建仓库

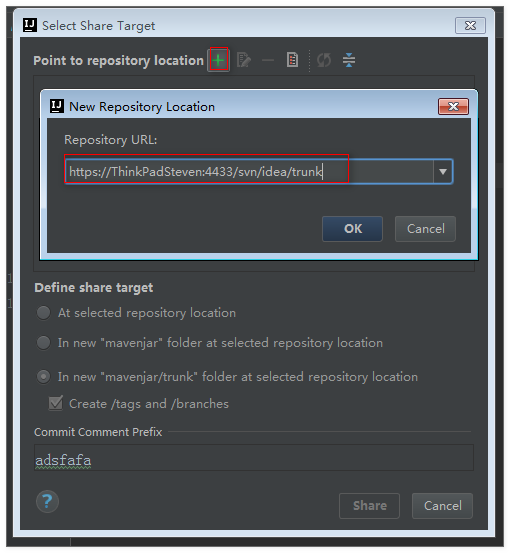


### 上传代码

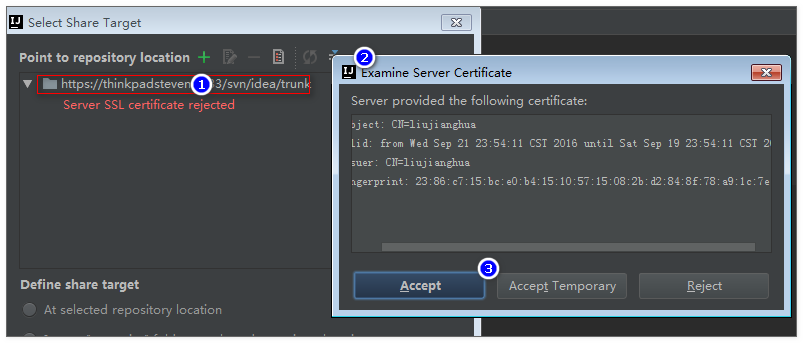
先添加忽略。( 前面的添加忽略文件)

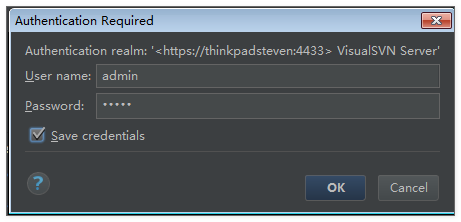


### 输入svn服务器地址

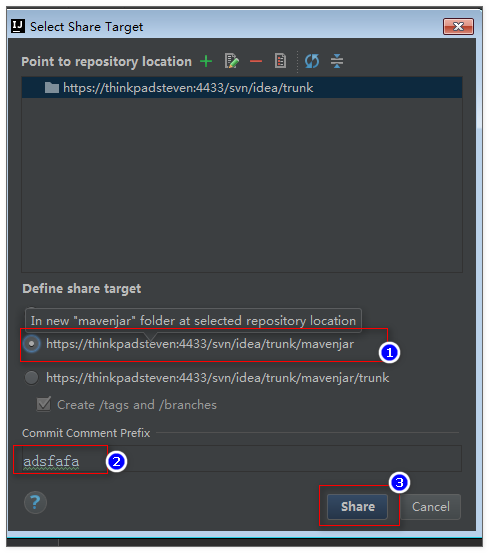


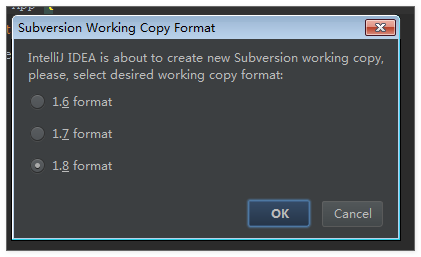
输入密码：





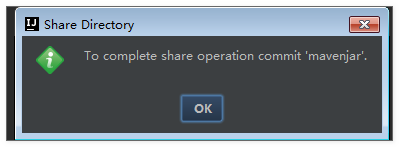
### 选择创建目录结构



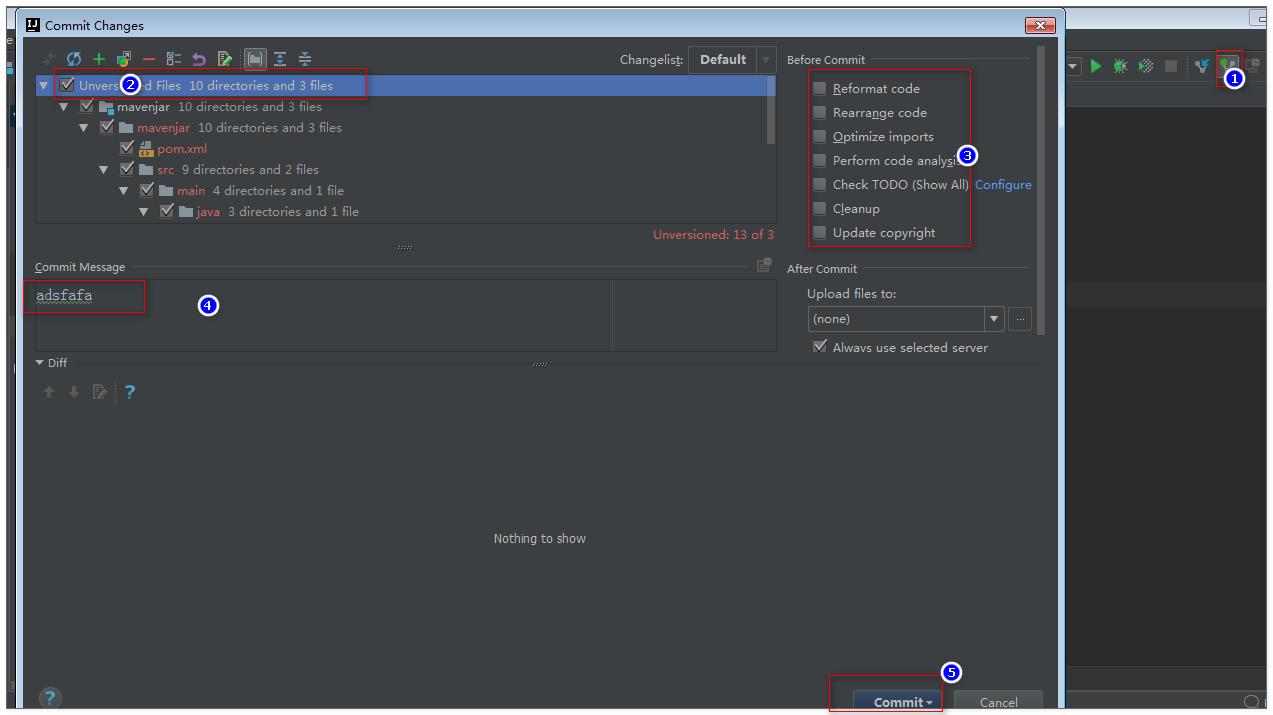


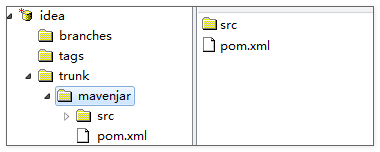
### commit

成功提示

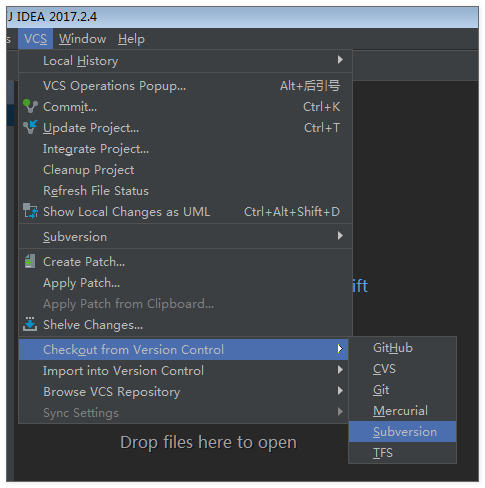


commit



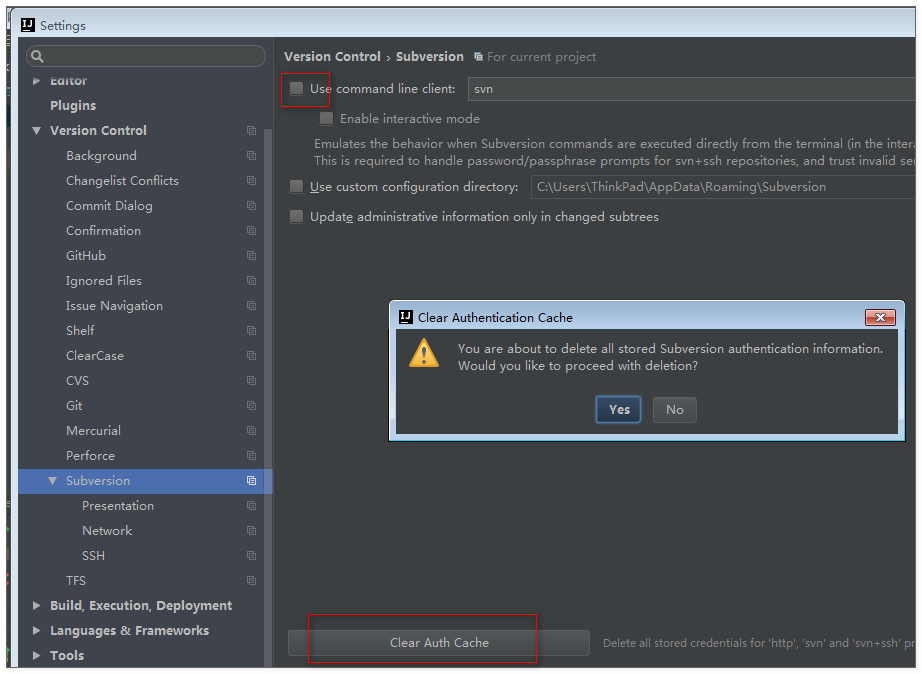


## 从SVN服务器上下载代码



## SVN的注意事项

当用户名和密码不正确时，需要清除原来存储在项目中的用户名和密码。



# Git集成idea的使用

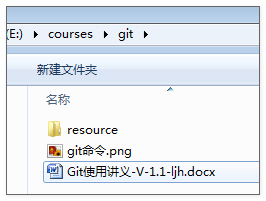
## 集成Git环境

### 安装git服务器

这里使用github作为远程git服务器。

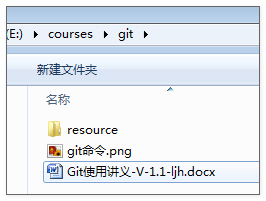
### 在本地安装git服务器

参考资料中安装本地git服务器章节。

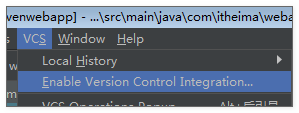


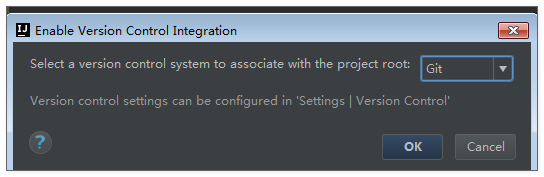
### 在本地安装git小乌龟

参考资料安装小乌龟

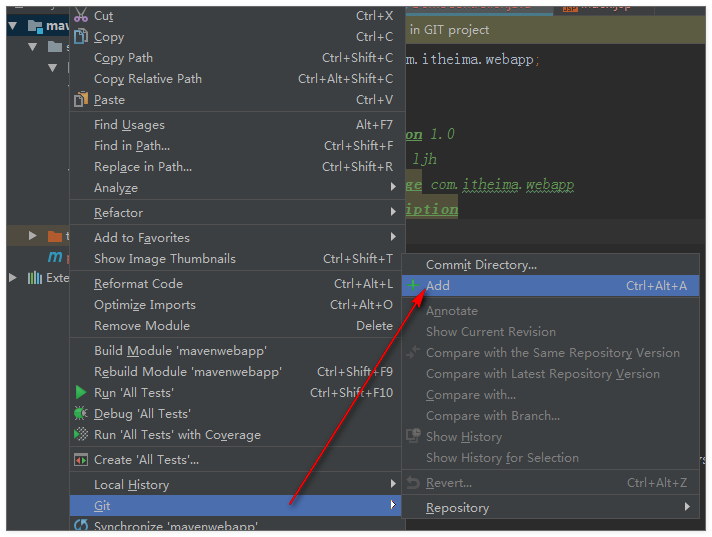


## 开启版本控制

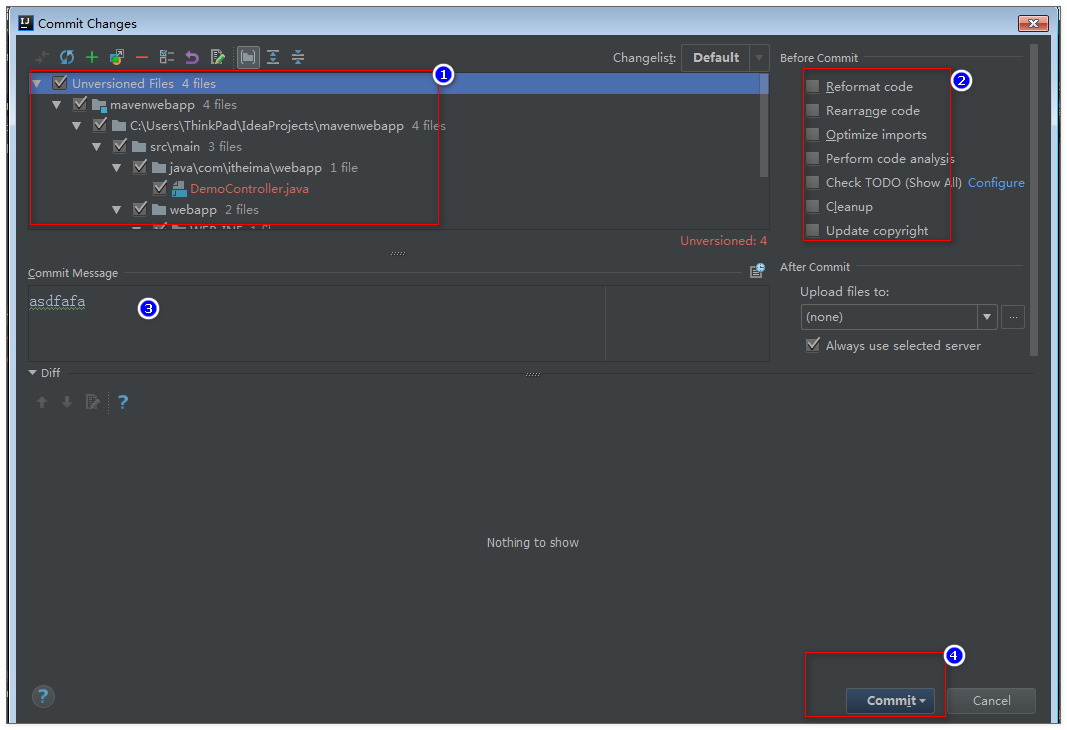




## git add

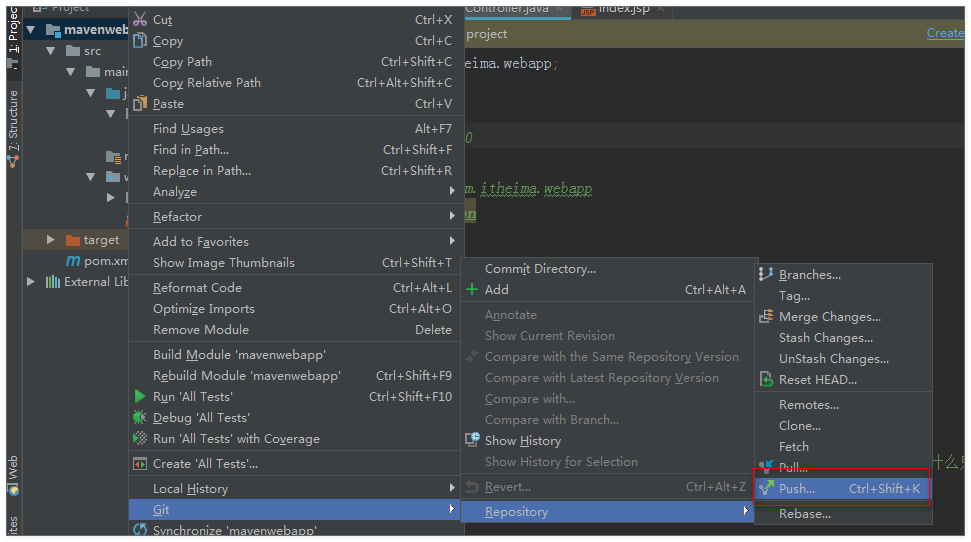


## git commit



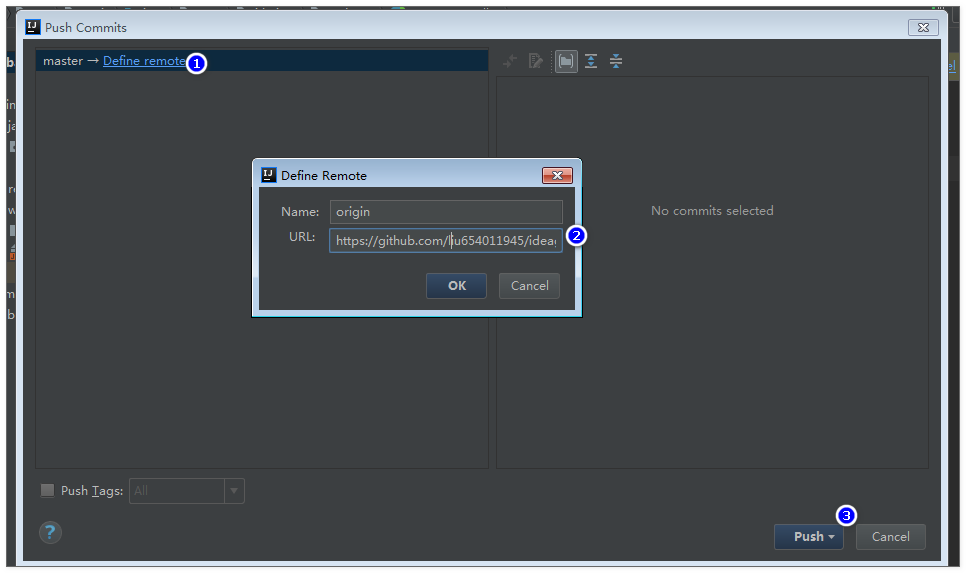
## git push

git push 推送需要推送代码到远程仓库，这里使用github作为远程仓库。

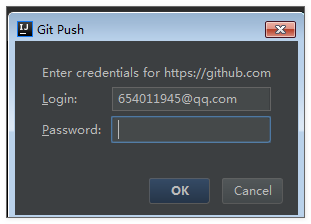


## 定义远程连接

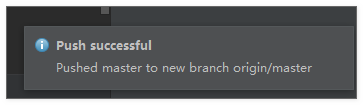
https://github.com/liu654011945/ideagittest



## 提示密码，输入密码

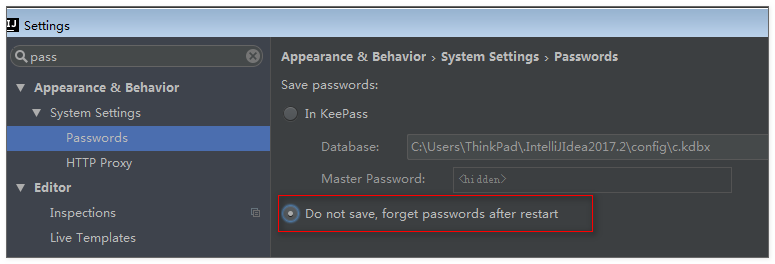


输入密码即可。



## git使用注意事项

如果出现问题，先清理密码：



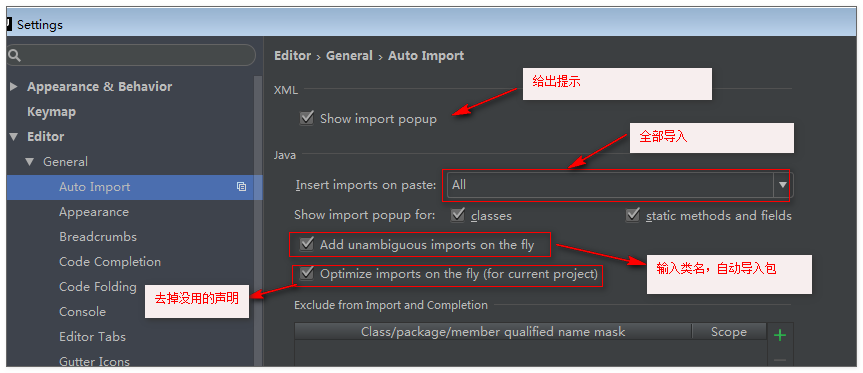
# debug调试

点击即可

# 使用技巧及快捷键的使用

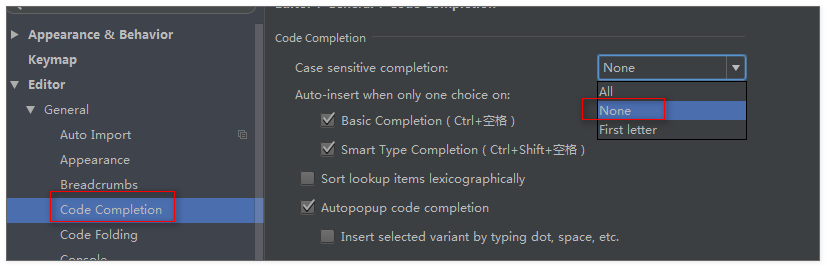
## 常见使用技巧

### 设置代码自动导入



### 设置代码输入从任意字母都能提示

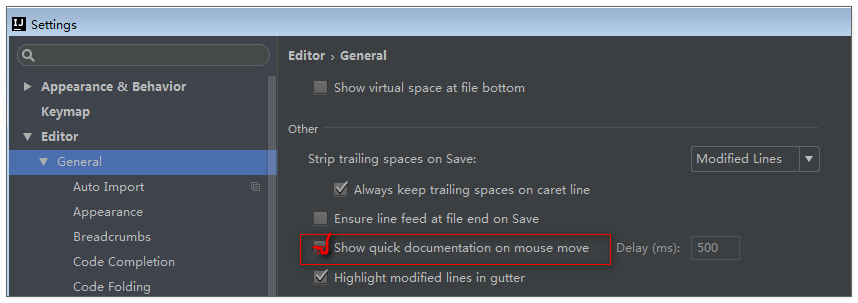
默认的情况下输入代码时，都是从首字母提示的，所以可以改为：任意提示



### 设置代码提示

在idea中没有类似于 eclipse中鼠标移动到方法上边就可以查看方法参数的功能。

需要开启，如图：

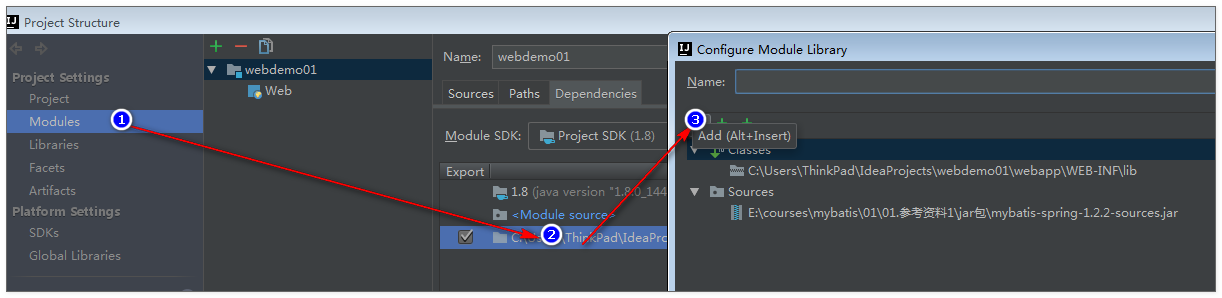


一般建议不开启，要看参数及注释：

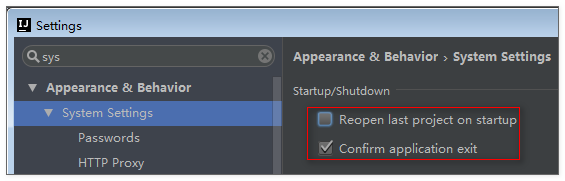
查看参数： CTR +P

查看注释：CTR+Q

### 查看源码



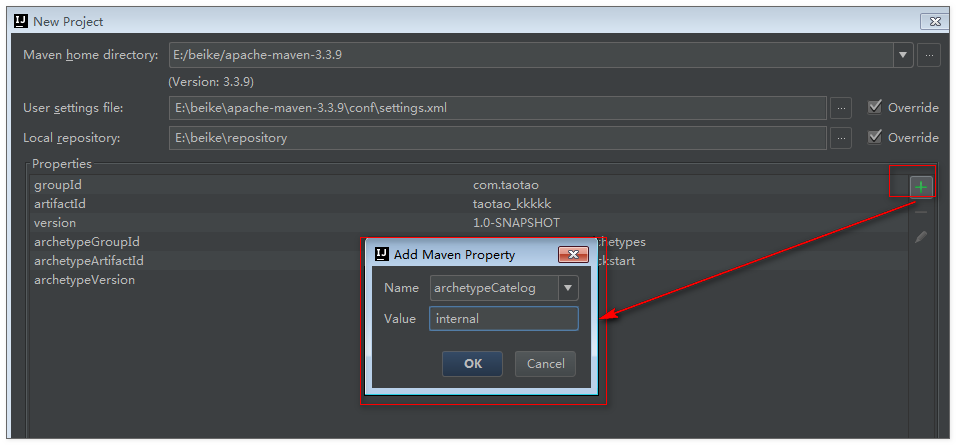
### 设置自定义选择打开项目



### 解决maven项目创建打开项目时慢的问题

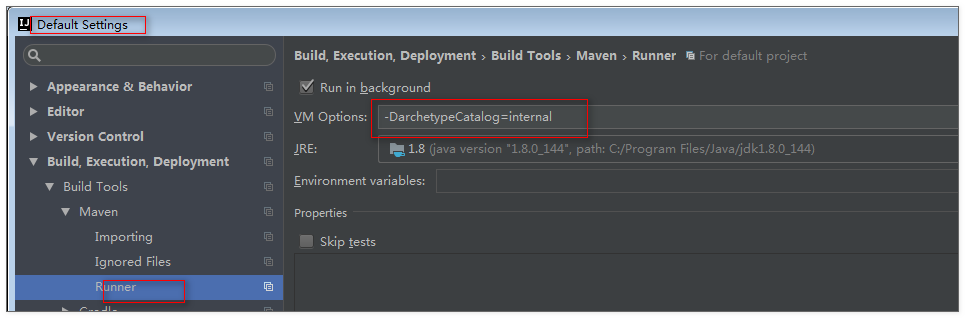
第一种：在每次创建maven项目的时候输入一个配置：

archetypeCatalog=internal



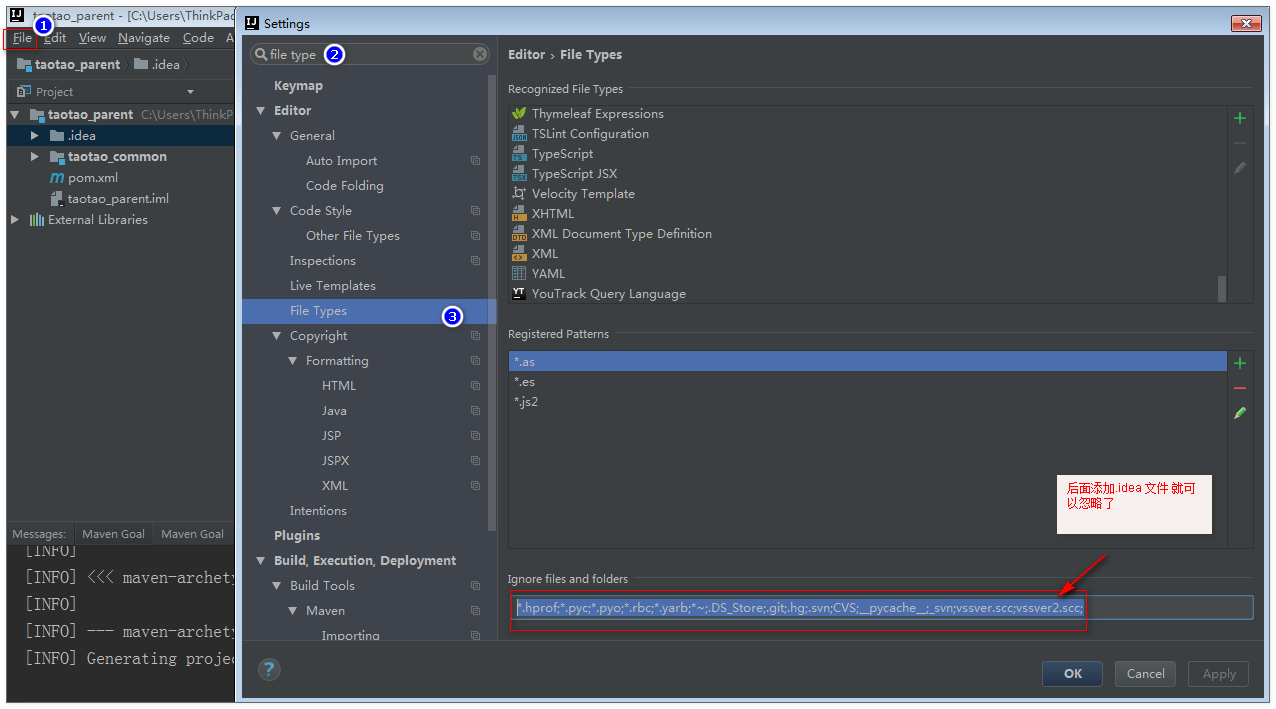
第二种：配置如下即可

-DarchetypeCatalog=internal



### 忽略.idea等文件

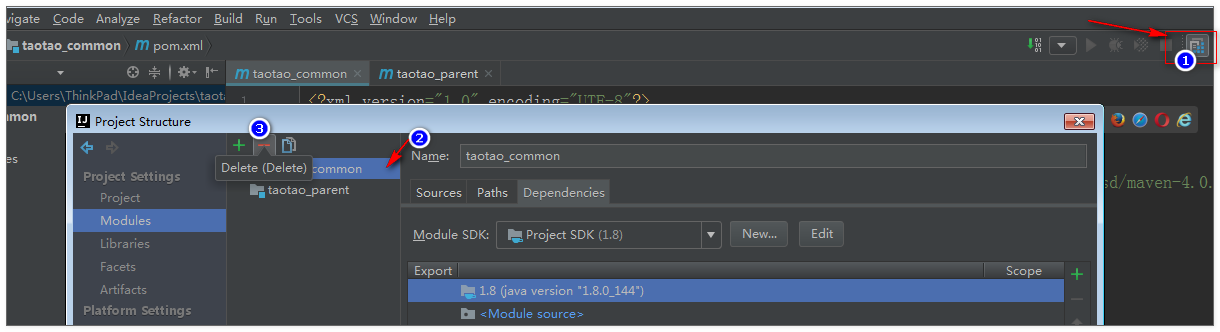
idea 中，有.idea 和.iml等文件，类似于eclipse中的.classpath .project 文件，不需要我们关心。如果不想看见它，就需要设置一个忽略。这有助于我们开发管理。如下图。



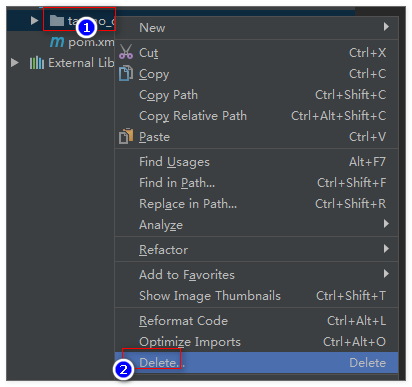
### 删除项目

idea的删除有些奇怪，需要先删除模块再删除目录结构。

当然也可以直接在windows系统中干掉目录文件夹。



再删除显示在项目栏中的项目。



此时右击鼠标，才会出现delete的按钮，不然是不会出现的。

再删除父模块中的模块配置，如图删除框中的模块。



## 快捷键盘

1.CTR+N 搜索类

2.CTR+SHIT+N 搜索文件

3.CTR+ALT+空格 代码提示（类似于ALT+/）

4.ALT+F7 查询在某处使用

5.CTR+Q 查看代码提示及内容 CTR+P 查看代码中的参数

6.CTR+B 查看类的声明 还可以直接按住：CTR 再点击鼠标 （和eclipse的用法是一样的）

7.CTR+F12 查看文件列表（File Structure）

8.SHIFT+F6 重命名（按住ENTER）重命名成功。

12 ALT+INSERT :生成getter和setter的方法

14 CTR+ALT+T 选中某一个段代码进行try catch

16 CTR+ALT+B ：查看接口或者抽象类的子类。

17.CTR+SHIT+/ :注释（整体注释）

18.CTR+/ :行注释

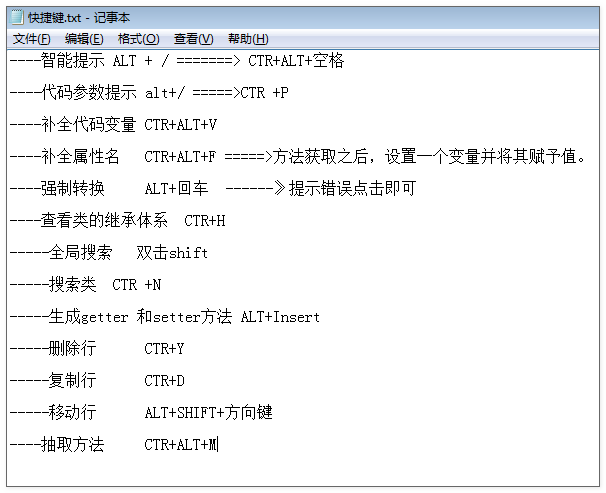
19.CTR+D :换行 并复制

20TR+Y:删除行

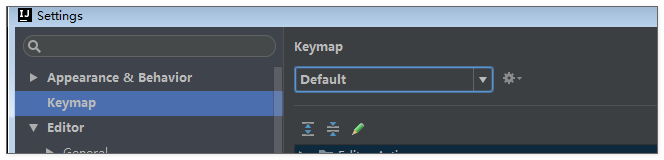
ALT + ENTER

//以下是比较有用的几个快捷键

http://blog.csdn.net/dc\_726/article/details/42784275



# 快捷键使用eclipse的配置



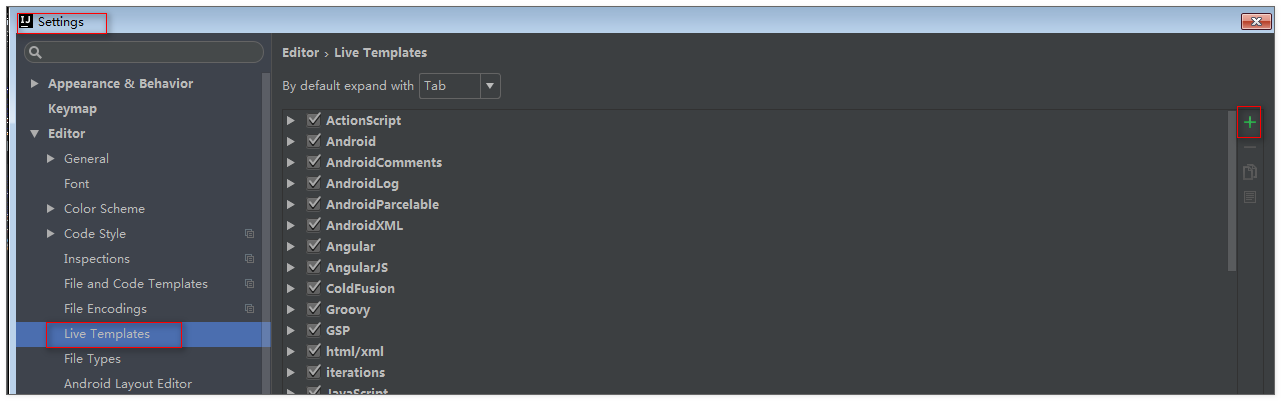
可以不使用idea自动的快捷键 ，可以使用eclipse的快捷键。

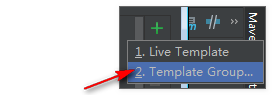
# 动态自定义模板

比如，在某种情况下，需要自定义某些模板，那么就可以在idea中方便实现。

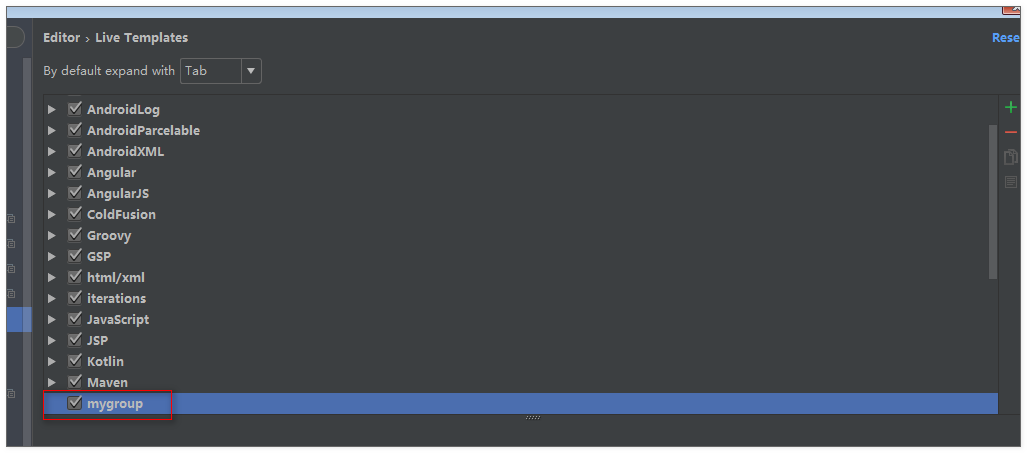
## 自定义代码提示syso

添加一个组：

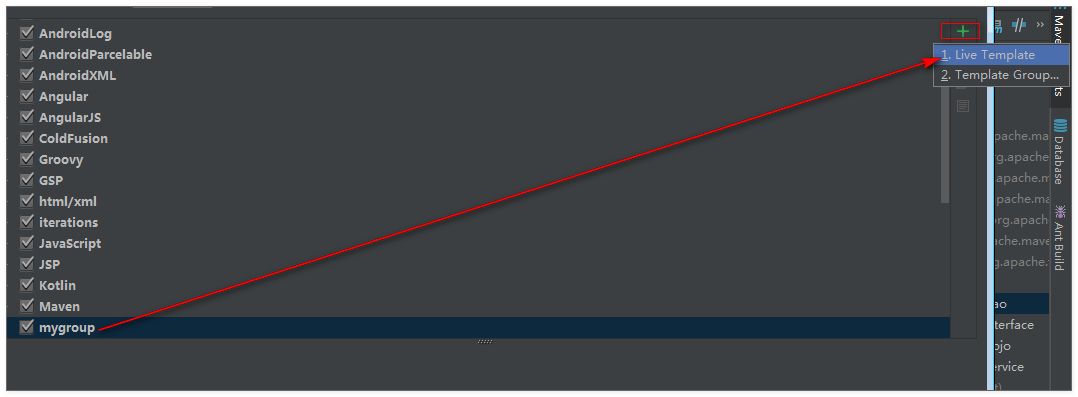




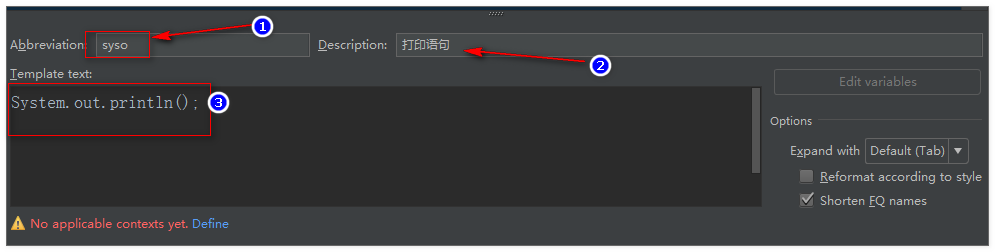
命名为mygroup 确定，之后创建了一个组：



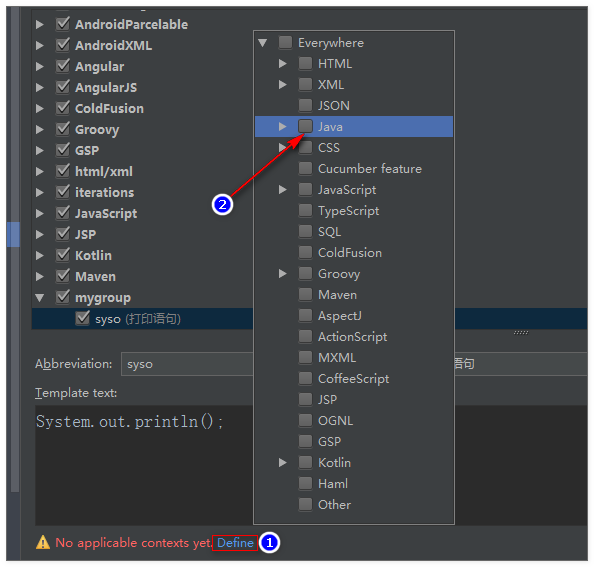
为这个组添加模板：



添加打印快捷键（syso）

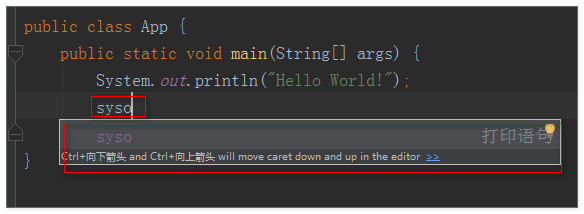


添加定义：

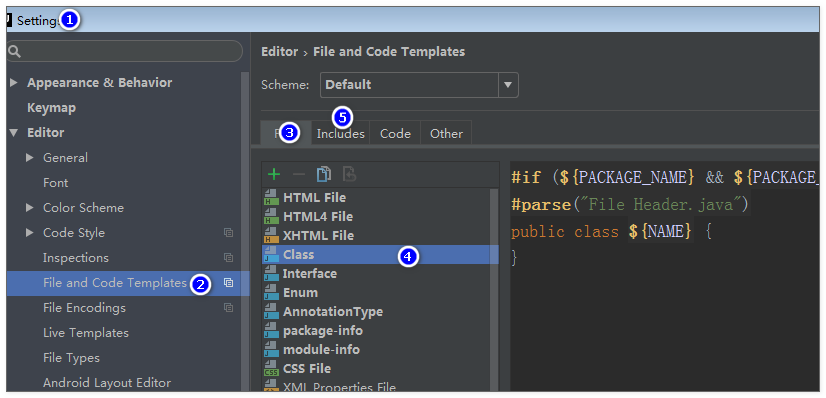


点击OK 即可

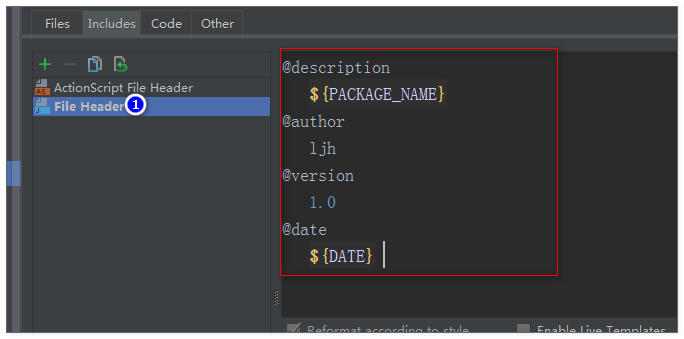
在代码中测试如下，输入syso 提示如下：



## 设置创建类的模板注释



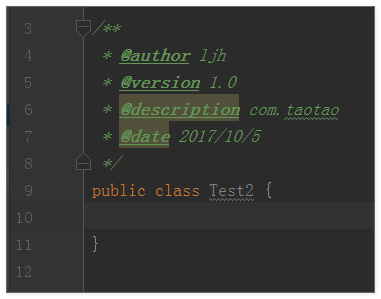
设置注释：



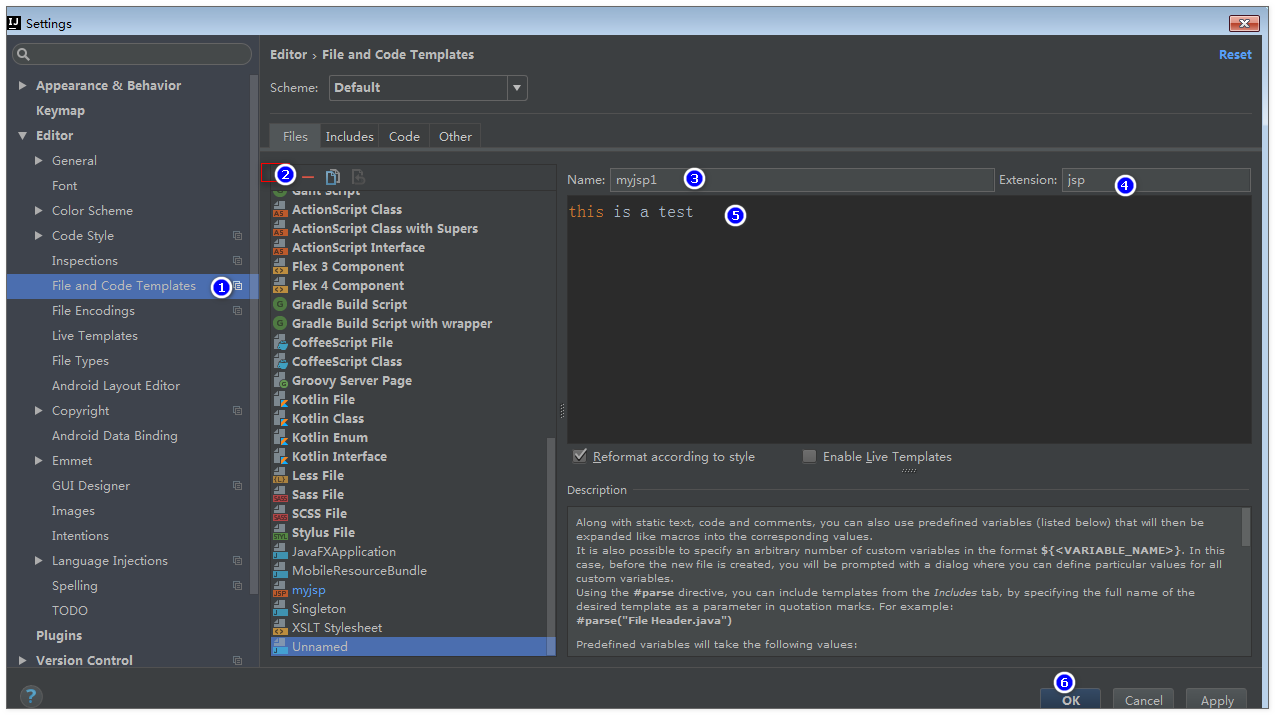
里面的内容如下:

|  |
| --- |
| /\*\*  \* @description  **${**PACKAGE\_NAME**}** @author  ljh @version  1.0 @date  **${**DATE**}** \*/ |

点击OK.创建类测试效果:

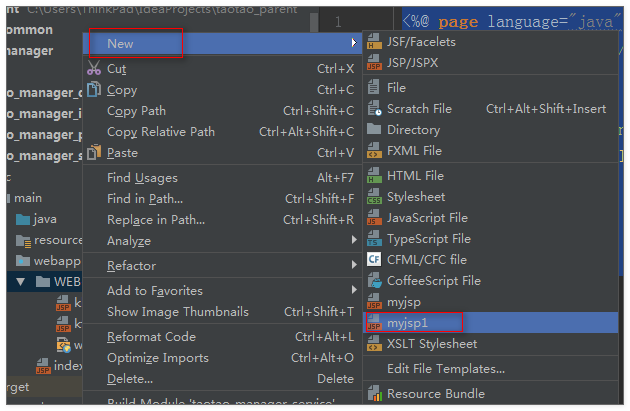


## 自定义jsp



如图:自定义jsp模板,

测试如下:



# 学习的视频及文档资料

更多官方学习信息

IntelliJ IDEA 主要特性介绍 1：https://www.jetbrains.com/idea/features/

IntelliJ IDEA 主要特性介绍 2：

https://www.jetbrains.com/idea/features/editions\_comparison\_matrix.html

官方快速入门：http://confluence.jetbrains.com/display/IntelliJIDEA/Quick+Start

官方在线帮助文档：http://www.jetbrains.com/idea/webhelp/getting-help.html

官方 wiki：http://wiki.jetbrains.net/intellij

更多官方资讯跟踪途径

官方博客：http://blog.jetbrains.com/idea/

IntelliJ IDEA 官方 community：https://intellij-support.jetbrains.com/hc/en-us/community/topics

IntelliJ IDEA 官方 issue：https://youtrack.jetbrains.com/issues/IDEA

YouTube：https://www.youtube.com/user/intellijideavideo

Twitter：https://twitter.com/IntelliJIDEA

Facebook：https://www.facebook.com/IntelliJIDEA

JetBrains 新浪微博：http://www.weibo.com/u/3220313942

JetBrains Google+：https://plus.google.com/+jetbrains

IntelliJ IDEA Google+：https://plus.google.com/+intellijidea

github上的文档：

https://github.com/judasn/IntelliJ-IDEA-Tutorial

//常用的插件的使用安装。

http://blog.csdn.net/sunny243788557/article/details/26556967