## Maven下载安装和配置

Maven是apache的一个项目，用java开发的，需要依赖JDK

<http://maven.apache.org/download.cgi>

到官网下载

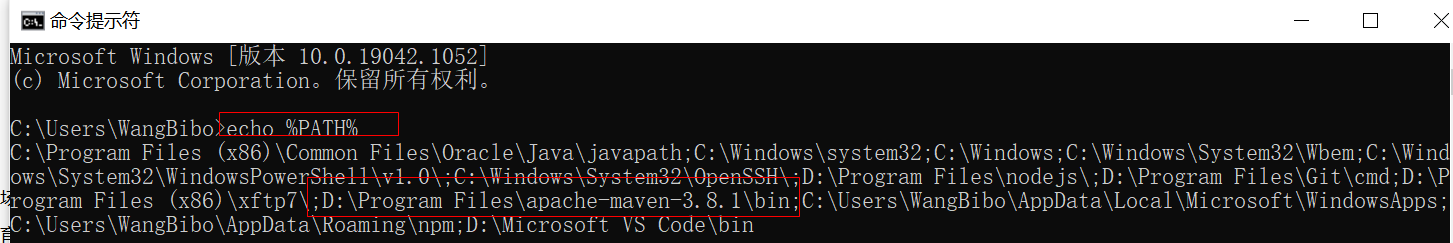


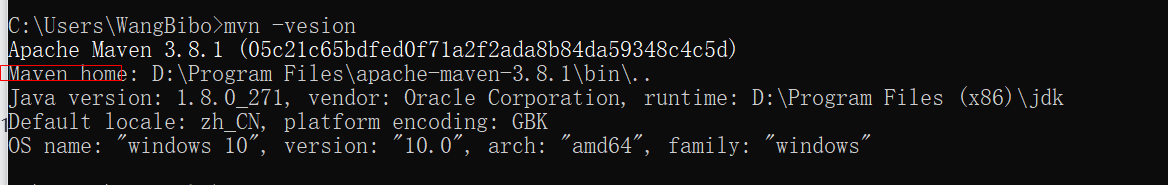
PS:上面标注的Linux不要介意。

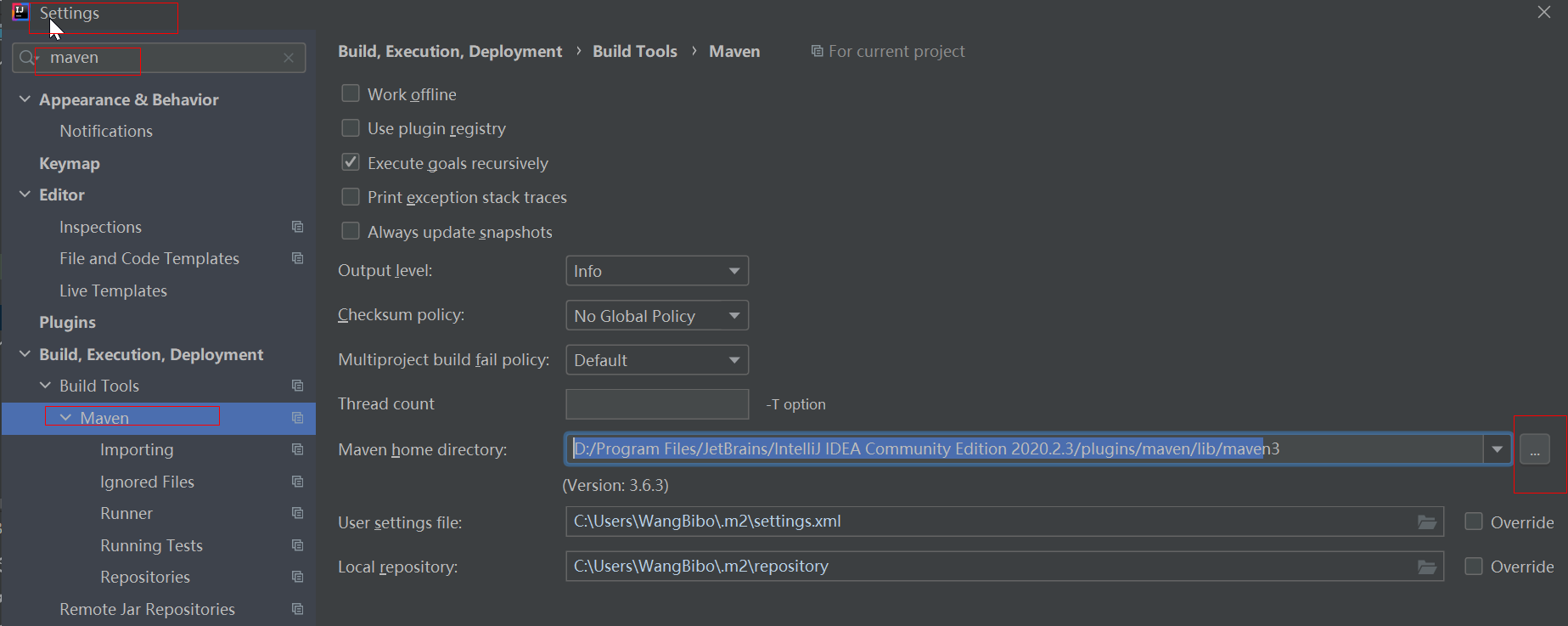
M2\_HOME=D:\Program Files\apache-maven-3.8.1 注意没有bin

M2=%M2\_HOME%\bin

MAVEN\_OPTS=-Xms256m -Xmx512m







参考：<https://www.cnblogs.com/stars-one/p/10958796.html>

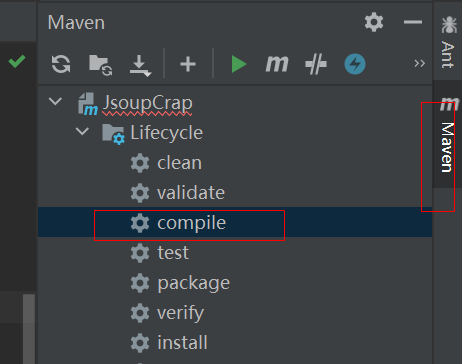
Maven的settings.xml文件配置，配置仓库镜像和本地仓库

|  |
| --- |
| <settings>  <mirrors>  <mirror>  <id>alimaven</id>  <name>aliyun maven</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  <mirrorOf>central</mirrorOf>  </mirror> <mirror>  <id>repo2</id>  <mirrorOf>central</mirrorOf>  <name>Human Readable Name for this Mirror.</name>  <url>http://repo2.maven.org/maven2/</url>  </mirror>  </mirrors>  <localRepository>D:\Program Files\mavenrepo</localRepository>  </settings> |

把C:\Users\WangBibo\.m2下面的settings.xml替换了

配置了阿里云镜像后，主流的包主流版本都能下载。

查询https://mvnrepository.com/ 依赖maven配置,如何查询阿里云maven仓库上的软件版本

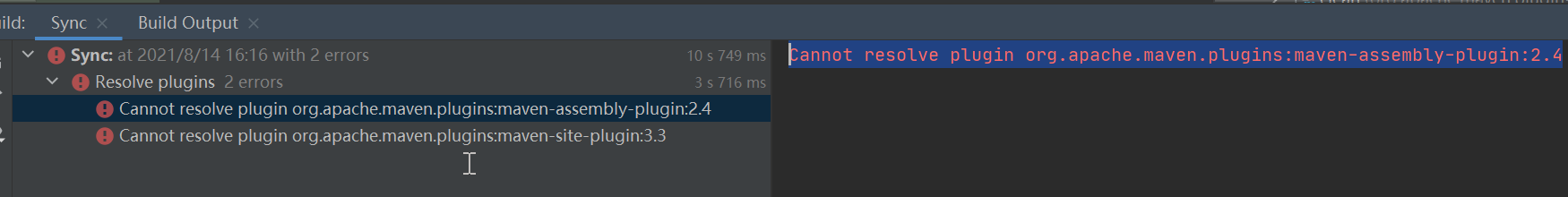


**问题：Cannot resolve plugin org.apache.maven.plugins:maven-install-plugin:2.4**

org/apache/maven/plugins/目录下  
把所有的文件删除掉，再次执行更新就能够解决这个问题。如果还是同样的问题，那么你就必须考虑

1. 网络问题  
   是否需要配置代理  
   2.仓库问题  
   是否仓库有问题。

**问题：不能获取xx**



从 mvnrepository 查询发现存在 3.8.2 的版本，

[https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-site-plugin](https://link.ld246.com/forward?goto=https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-site-plugin" \t "_blank)

于是把

<!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-site-plugin -->

<dependency>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-site-plugin</artifactId>

<version>3.8.2</version>

</dependency>

加入 pom.xml 把 jar 文件拉下来，解决。

**→问题：不能访问https://mvnrepository.com/artifact/，授权失败**

更改idea中的settings设置，解决证书等问题

向File - settings中的maven - importing中添加这句

-Dmaven.wagon.http.ssl.insecure=true -Dmaven.wagon.http.ssl.allowall=true

能够解决证书问题，如图所示：

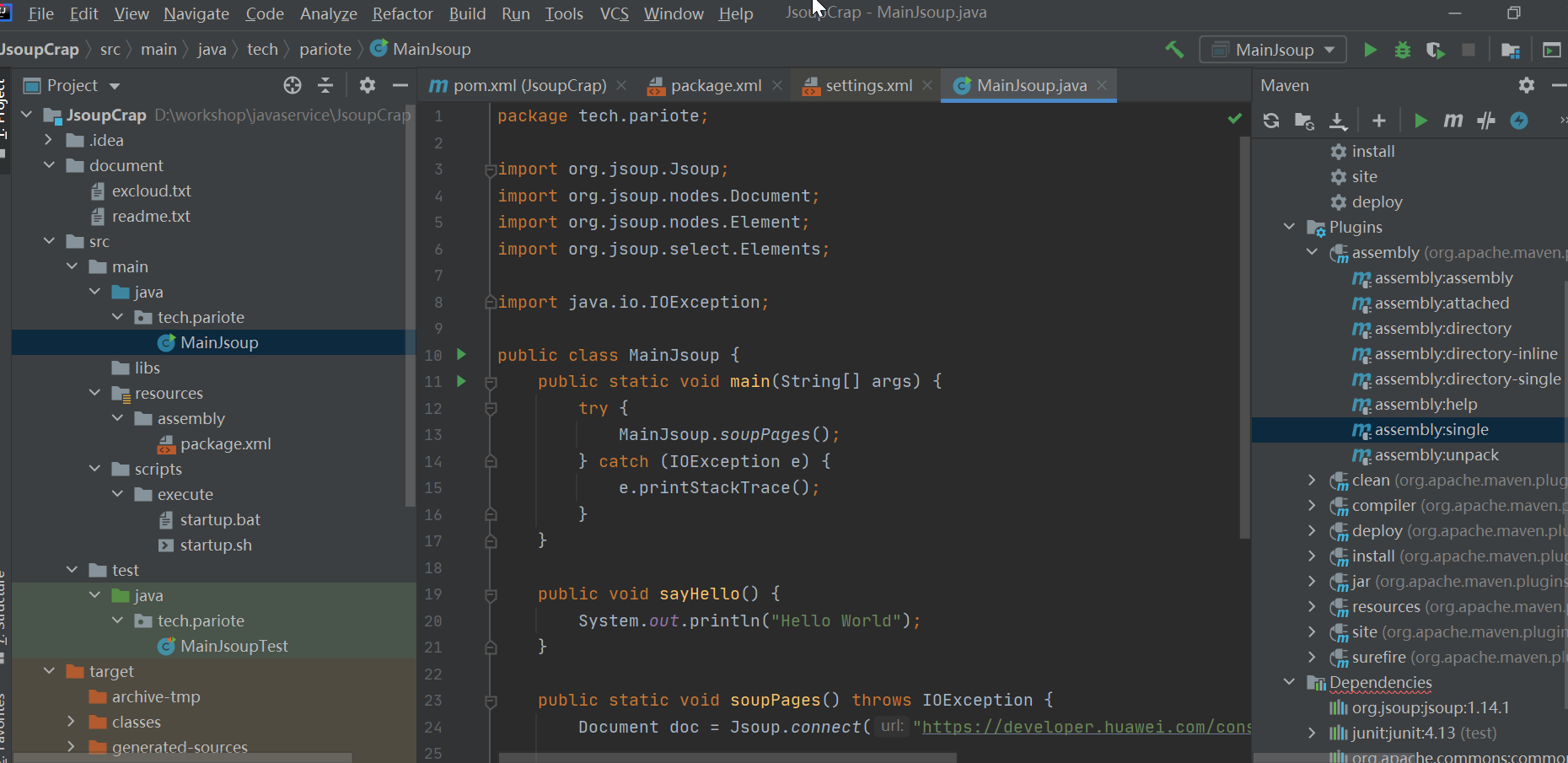
Article: 彻底解决Could not transfer artifact org.apache.maven.plugins问题

<https://blog.csdn.net/weixin_44005516/article/details/108293228>

## 普通java工程打包成可以直接运行的zip包

工程目录

D:\workshop\javaservice\JsoupCrap



工程githup上归档

https://github.com/wangbiboseu/JsoupCrap

核心plugin:

maven-assembly-plugin

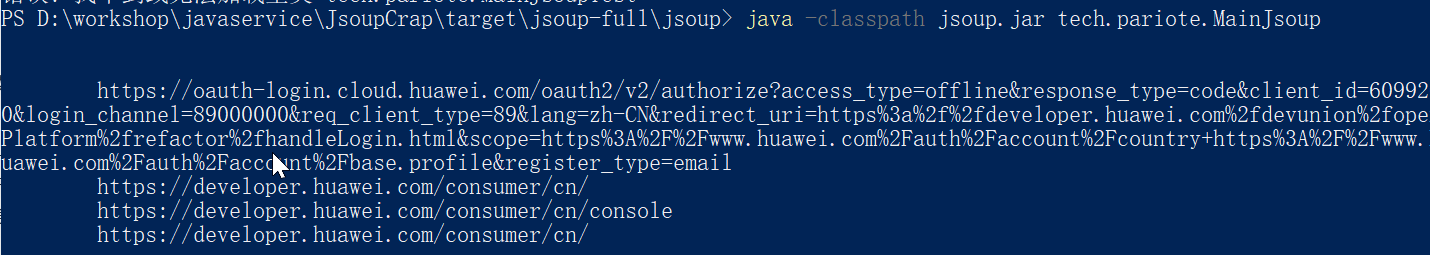
maven-jar-plugin

<https://blog.csdn.net/daerzei/article/details/82883472>

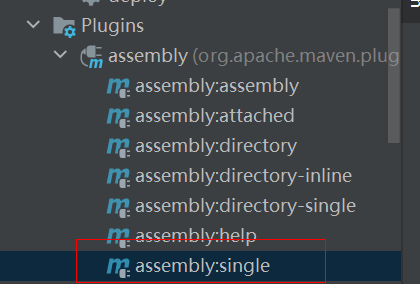
-->测试

java -classpath jsoup.jar tech.pariote.MainJsoup

执行成功



**问题：测试打zip包错误**



Failed to execute goal org.apache.maven.plugins:maven-assembly-plugin:2.4:single (default-cli) on project JsoupCrap: Error reading assemblies: Error reading descriptor: src/main/resources/assembly/package.xml: Unrecognised tag: 'directoryl' (position: START\_TAG seen ...<fileSet>\r\n <directoryl>... @42:25)

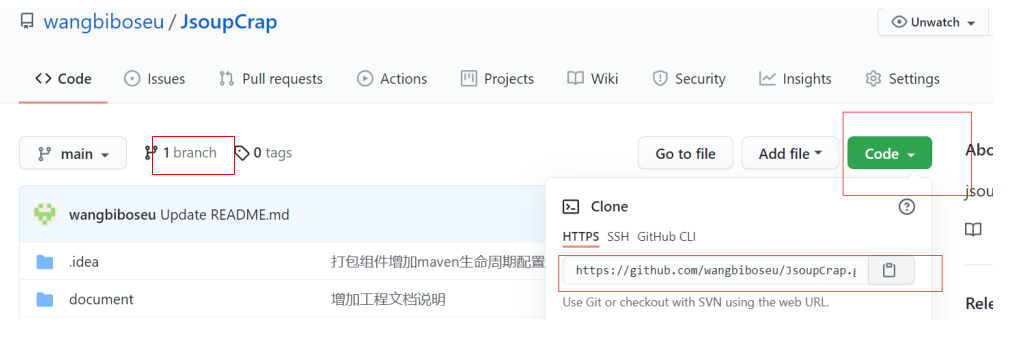
标签拼写错误

## Github提交

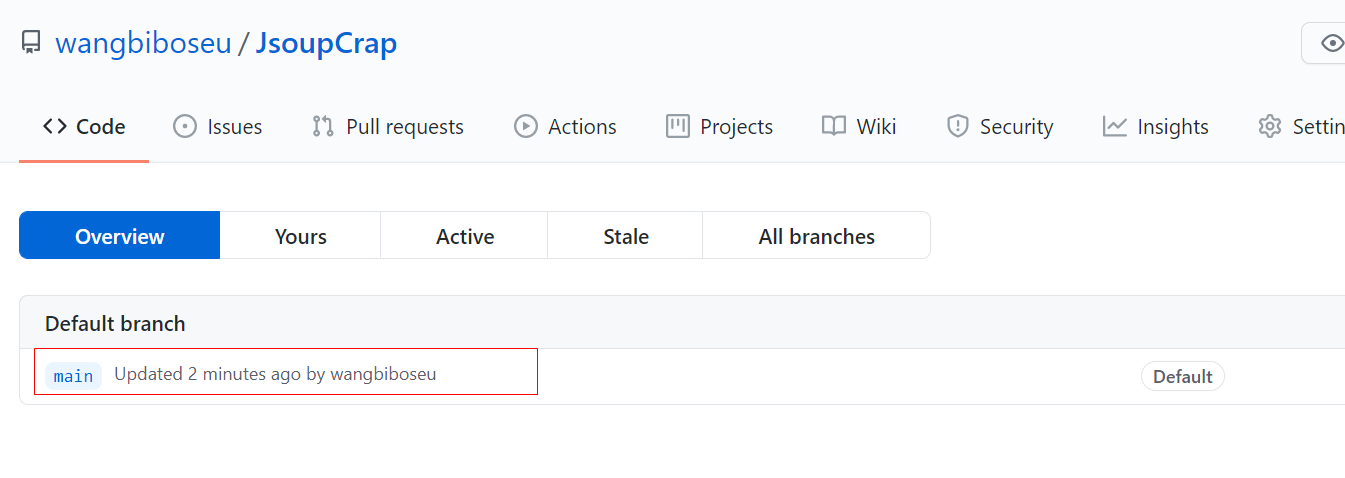
接下来依次输入以下代码即可完成其他剩余操作：

1. **下载一个项目工程**

从github上获取下载链接



获取分支名称



在git客户端执行命令，下载工程

git clone https://github.com/wangbiboseu/JsoupCrap.git -b main

**2、提交代码**

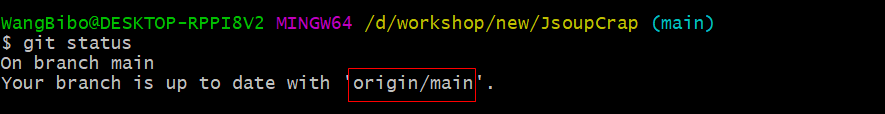
git add \*    （注：别忘记后面的.，此操作是把Test文件夹下面的文件都添加进来）

git commit  -m  "提交信息"  （注：“提交信息”里面换成你需要，如“first commit”）

git push -u origin main（注：此操作目的是把本地仓库push到github上面，此步骤需要你输入帐号和密码)

main为分支名称，

orgin和main什么概念？



**3、给项目添加.gitignore**

.gitignore可以忽略你不想上传的文件，比如doc,target,classes等等

只需要在.git同目录下新增.gitignore文件，然后添加不需要上次的目录即可，该文件无需提交到git仓库，是本地git配置文件，里面路径下的文件git自动忽略存在即变化，比如

/doc/

/target/

/.idea/

/offer.iml

article：git中增加.gitignore

<https://www.jianshu.com/p/a49124700abc>

**4、修改READ.ME文档**

支持在线修改说明，支持上传截图，支持markdown便捷器



并且可以贴图