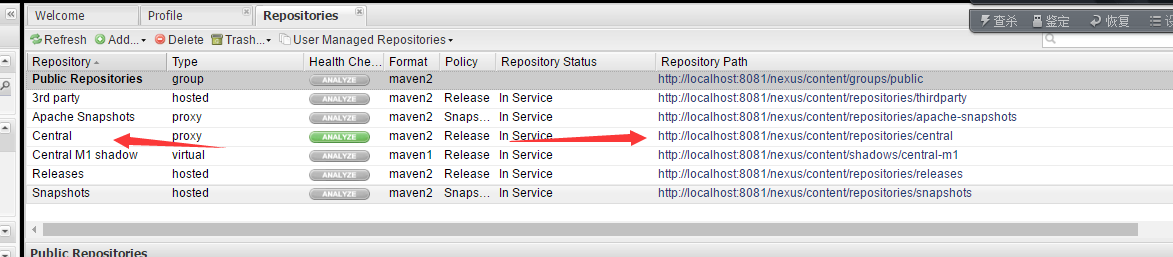
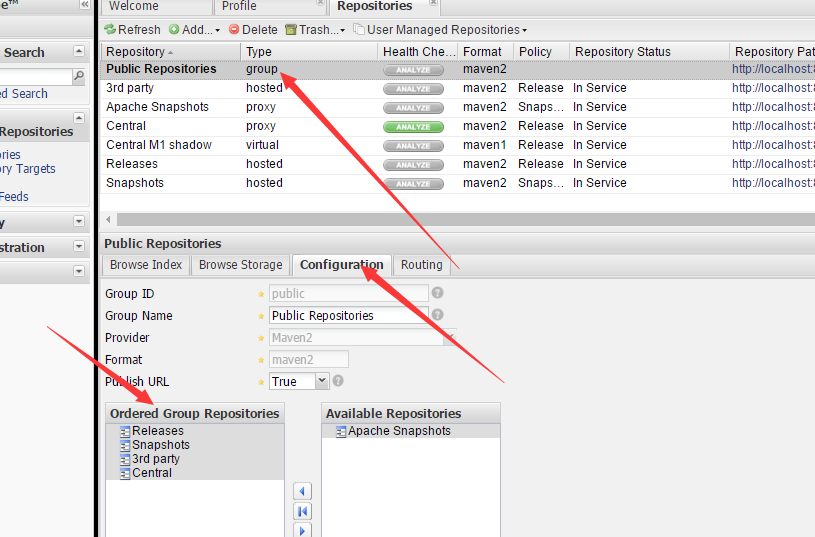
# 1、实现从私服下载（依赖会两边各放）

## 1、添加仓库，从中央工厂（本地）下载



|  |
| --- |
| <repositories>  <repository>  <id>nexus</id>  <name>Nexus Repository</name>  <!-- 18、如果从中央工厂下载 -->  <url> http://localhost:8081/nexus/content/repositories/central/</url>  </repository>  如果想添加多个工厂（snapshotd等）url，则在这里再新建repository，再添加url，但这样比较麻烦，所以看2  </repositories> |

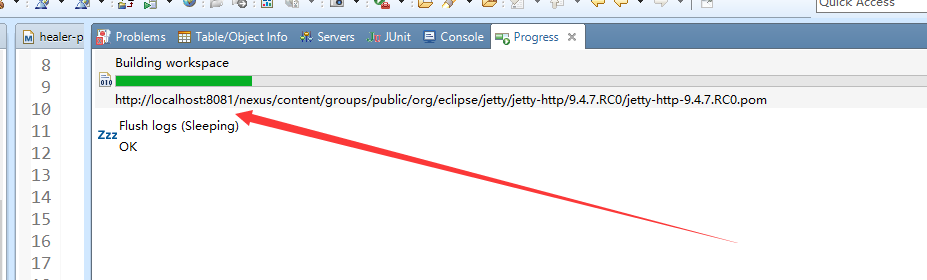
## 2、一个url解决问题，也就是用到了上面的类型中的group，只要下面的config配置中存在其他的工厂，复制这个group的url即可（这里的地址如果放到谁的ip则会从谁的工厂下载，这就是我们所需要的将来的私服）



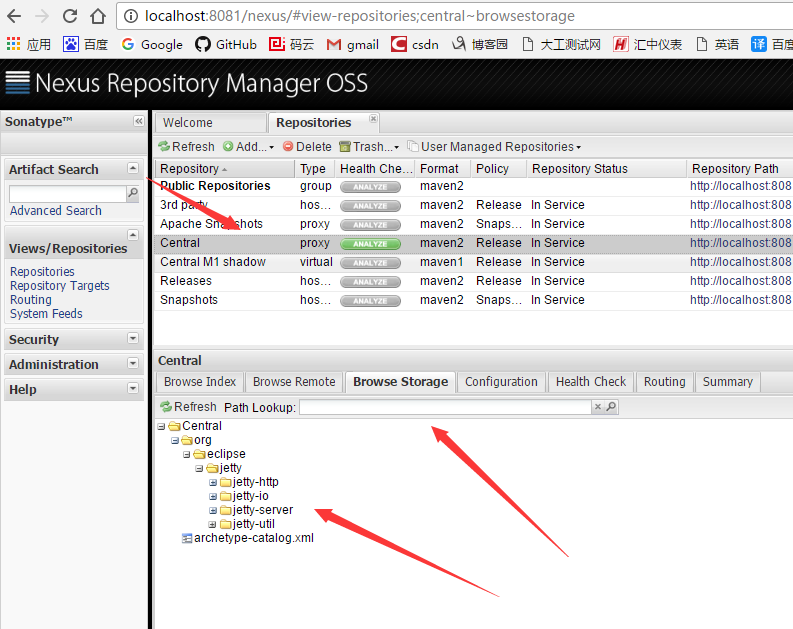
|  |
| --- |
| <repositories>  <repository>  <id>nexus</id>  <name>Nexus Repository</name>  <url>http://localhost:8081/nexus/content/groups/public/</url>  </repository>  </repositories> |

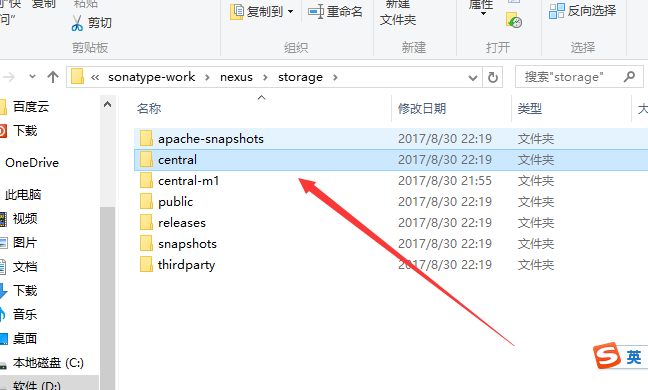
### 1、测试 parent添加一个jetty的依赖，会发现从以上地址中开始下载

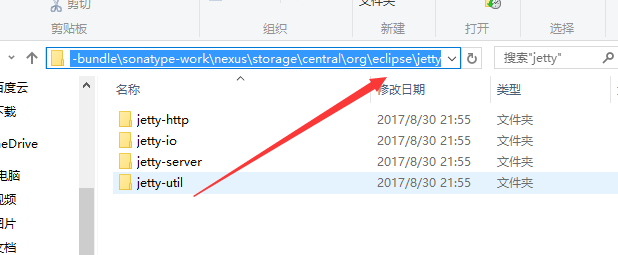
|  |
| --- |
| <dependency>  <groupId>org.eclipse.jetty</groupId>  <artifactId>jetty-server</artifactId>  <version>9.4.7.RC0</version>  </dependency> |



### 2、打开浏览器，会发现中央工厂中有了依赖了







# 2、利用不同的开发环境使用不同的工厂，不更改代码

## 1、下面这个将来为了开发中能够依靠不用的开发环境而使用不同的url 地址（localhost会变化）仓库，为了不更改代码，所以配置到了maven的 server.xml的配置文件中去了。

|  |
| --- |
| <repositories>  <repository>  <id>nexus</id>  <name>Nexus Repository</name>  <url>http://localhost:8081/nexus/content/groups/public/</url>  <releases> <!-- releases 默认是开启的，这里true即可 -->  <enabled>true</enabled>  </releases>  <snapshots> <!-- snapshots默认是关闭的，需要手动开启 -->  <enabled>true</enabled>  </snapshots>  </repository>  </repositories> |

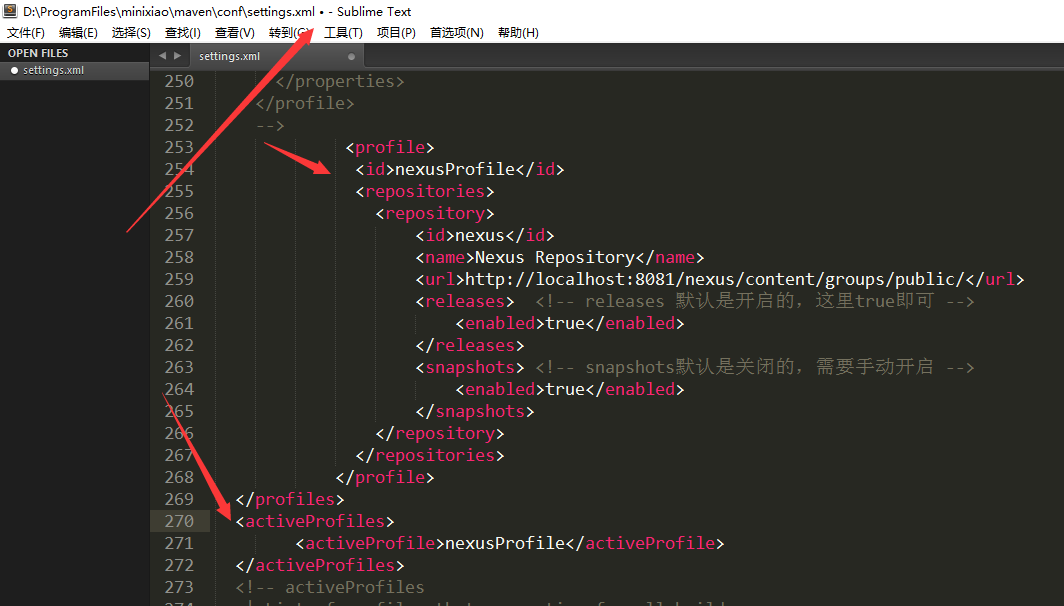
## 2、maven的setting.xml中添加如下（找到profiles的位置）

|  |
| --- |
| <profile>  <id>nexusProfile</id>  <repositories>  <repository>  <id>nexus</id>  <name>Nexus Repository</name>  <url>http://localhost:8081/nexus/content/groups/public/</url>  <releases> <!-- releases 默认是开启的，这里true即可 -->  <enabled>true</enabled>  </releases>  <snapshots> <!-- snapshots默认是关闭的，需要手动开启 -->  <enabled>true</enabled>  </snapshots>  </repository>  </repositories>  </profile> |

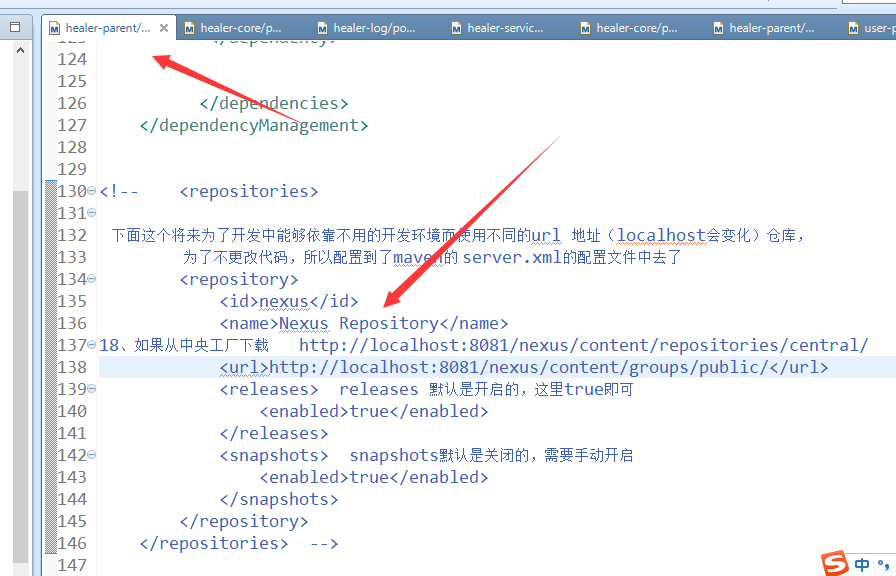
## 3、maven的setting.xml还需要添加如下才能激活上面的设置

|  |
| --- |
| <activeProfiles>  <activeProfile>nexusProfile</activeProfile>  </activeProfiles> |

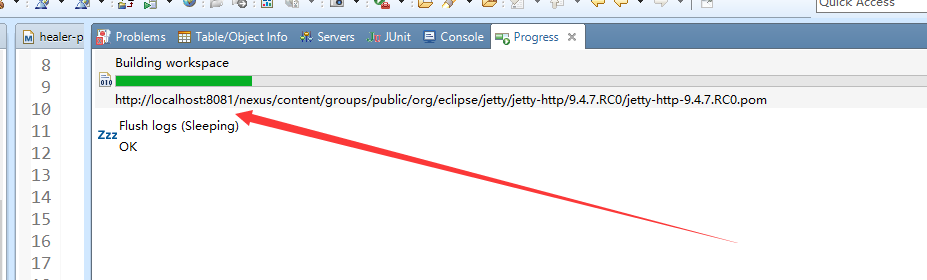
## 4、具体图片如下



## 5、然后将parent中pom.xml的以上配置删除

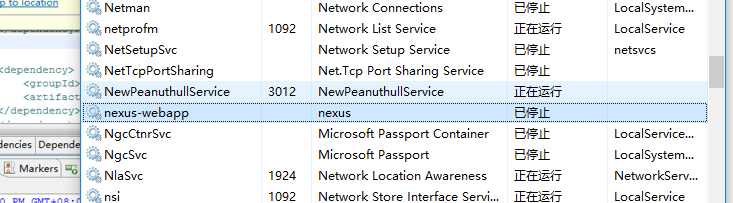


## 6、再次测试，成功

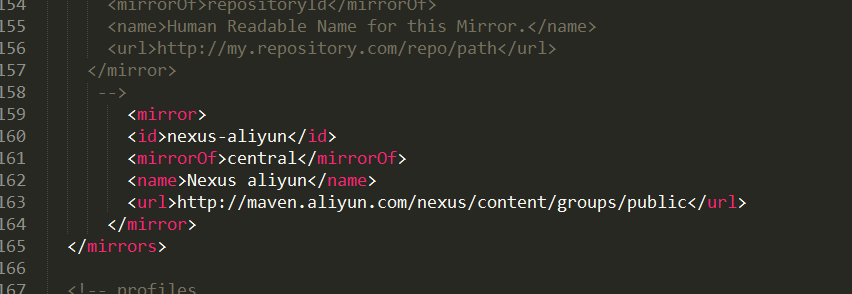


## 7、如果nexus关闭之后，则先会从以上配置，也就是中央工厂中找，如果发现关闭，则会自动加载网络下载(我的已经设置为aliyun了)

### 1关闭nexus



### 2、从aliyun 镜像mirror这里下载



# 3、配置镜像访问，这样以后就会自动从这里下载了（关闭（注释）阿里云的先）

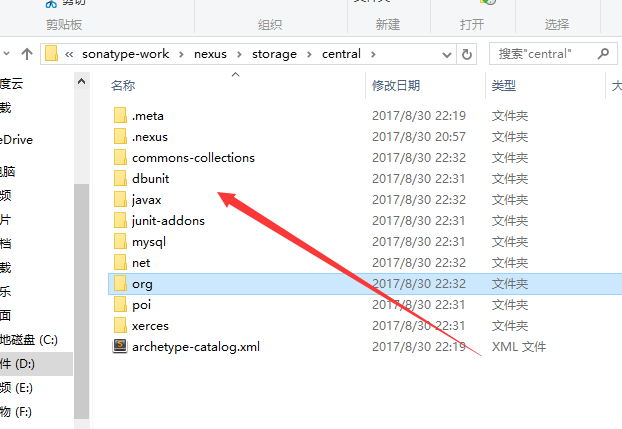
|  |
| --- |
| <mirror>  <id>nexus-MyNexus</id>  <mirrorOf>\*</mirrorOf> <!-- \*表示所有的工厂(snapshot等等)都来这里访问 -->  <name>Nexus MyNexus</name>  <url>http://localhost:8081/nexus/content/groups/public/</url>  </mirror> |

## 1、添加对snapshot的依赖的支持下载，添加一个profile，记得开激活profile，pom.xml会先从下面来找，这样snapshots设置了true，表示支持，但是路径确找不到，所以自动去上面的镜像中找了，名字和id都是随意按照个人喜好的

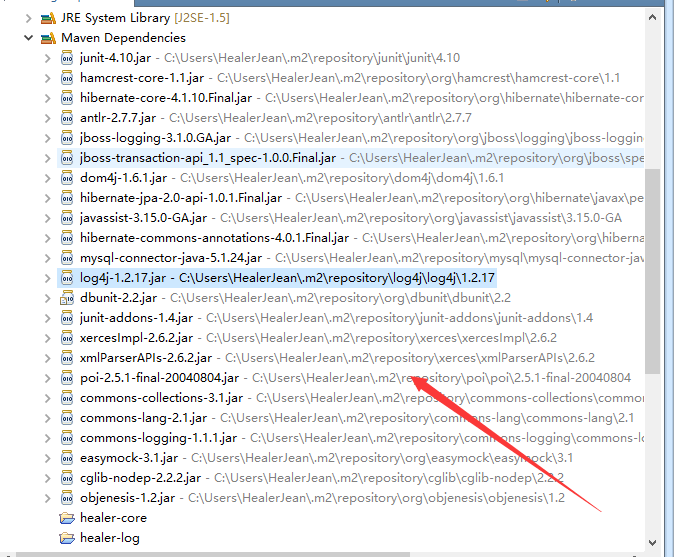
|  |
| --- |
| <profile>  <id>centralProfile</id>  <repositories>  <repository>  <id>central</id>  <name>Central Repository</name>  <url>http://\*</url>  <snapshots> <!-- snapshots默认是关闭的，需要手动开启 -->  <enabled>true</enabled>  </snapshots>  </repository>  </repositories>  </profile> |

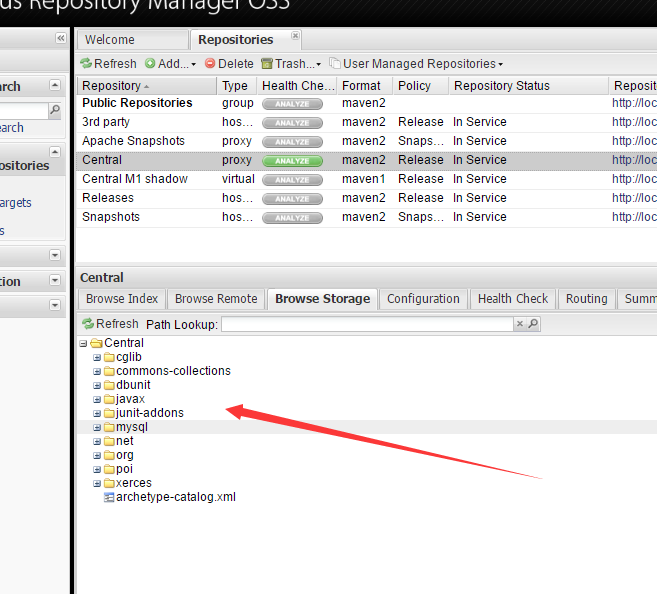
## 2、这个时候如果关闭了nexus，则不会下载依赖了

## 3、这个时候运行项目，会给我们的nexus仓库中下载依赖，因为我们上面的pom先会去上面的镜像（默认）中识别（不管m2本地中有没有） 先从nexus中找，找到有的话，在再去本地中找，也就是相当于nexus会帮助下载两份。



## 1、最终下编译所使用的是本地仓库的





# 4、解释[maven中snapshot快照库和release发布库的区别和作用](http://www.mzone.cc/article/277.html)

 maven中的仓库分为两种，

snapshot快照仓库和release发布仓库。

snapshot快照仓库用于保存开发过程中的不稳定版本，release正式仓库则是用来保存稳定的发行版本。定义一个组件/模块为快照版本，只需要在pom文件中在该模块的版本号后加上**-SNAPSHOT**即可(注意这里必须是大写)，如下：