[maven--私服的搭建（Nexus的使用）](http://blog.csdn.net/shenshen123jun/article/details/9084293" \t "_blank)和注意的问题

**私服是什么**

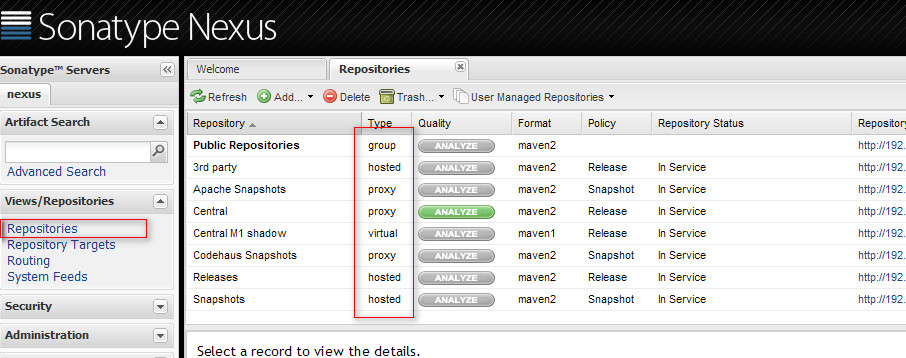
**私服，私有服务器，是公司内部Maven项目经常需要的东东，不总结一下，不足以体现出重视。Nexus是常用的私用Maven服务器，一般是公司内部使用。下载地址是**[**http://www.sonatype.org/nexus/go**](http://nexus.sonatype.org/downloads/)**。默认端口8081,这里我选择最新版nexus-2.5.0-04。**

**常用功能**

**Nexus常用功能就是：指定私服的中央地址、将自己的Maven项目指定到私服地址、从私服下载中央库的项目索引、从私服仓库下载依赖组件、将第三方项目jar上传到私服供其他项目组使用。**

**开启Nexus服务后访问url地址http://localhost:8081/nexus/(推荐使用自己的ip地址)，之后登录系统，用户名密码分别是：admin/admin123.**

**最频繁的就是点击左侧菜单栏的**[**Repositories**](http://127.0.0.1:8081/nexus/index.html)**按钮**

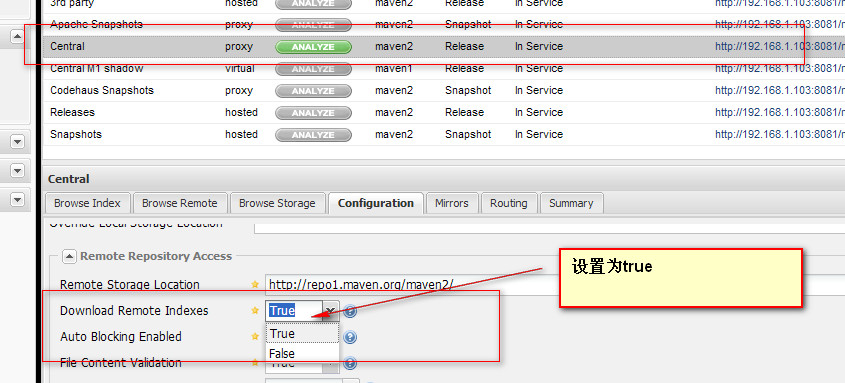


**一般用到的仓库种类是 hosted、proxy。Hosted代表宿主仓库，用来发布一些第三方不允许的组件，比如oracle驱动、比如商业软件jar包。Proxy代表代 理远程的仓库，最典型的就是Maven官方中央仓库、JBoss仓库等等。如果构建的Maven项目本地仓库没有依赖包，那么就会去这个代理站点去下载， 那么如果代理站点也没有此依赖包，就回去远程中央仓库下载依赖，这些中央仓库就是proxy。代理站点下载成功后再下载至本机。笔者认为，其实Maven 这个自带的默认仓库一般情况下已经够大多数项目使用了。特殊情况时在配置新的仓库，指定url即可，一般熟悉ExtJS的人操作这个Nexus都没什么问 题，单词不是很难，不明白的查查单词基本差不多。就是如果Sonatype公司对其做了国际化的处理就更好了。**

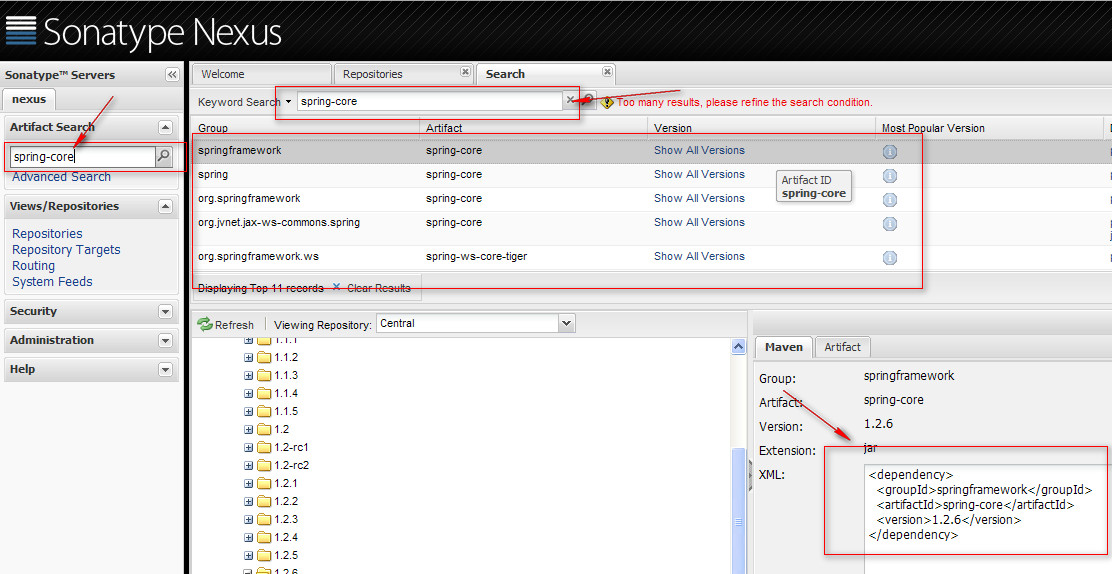
**hosted   类型的仓库，内部项目的发布仓库  
  
releases 内部的模块中release模块的发布仓库  
  
snapshots 发布内部的SNAPSHOT模块的仓库  
  
3rd party 第三方依赖的仓库，这个数据通常是由内部人员自行下载之后发布上去  
  
proxy   类型的仓库，从远程中央仓库中寻找数据的仓库  
  
group   类型的仓库，组仓库用来方便我们开发人员进行设置的仓库**

**maven项目索引**

**下载Maven项目索引，项目索引是为了使用者能够在私服站点查找依赖使用的功能**



**保存后后台会运行一个任务，点击菜单栏的**[**Scheduled Tasks**](http://127.0.0.1:8081/nexus/index.html)**选项即可看到有个任务在RUNNING。 下载完成后，Maven索引就可以使用了，在搜索栏输入要搜索的项，就可以查到相关的信息。例如spring-core**



**就可以检索出它的相关信息，包括怎么配置依赖信息。我们要想使用这个私服仓库，先在项目pom中配置相关私服信息**

**指定仓库**

**[html]** [view plaincopy](http://blog.csdn.net/shenshen123jun/article/details/9084293)

1. **<repositories>**
2. **<repository>**
3. **<id>**nexus**</id>**
4. **<name>**nexus**</name>**
5. **<url>**http://192.168.1.103:8081/nexus/content/groups/public/**</url>**
6. **<releases>**
7. **<enabled>**true**</enabled>**
8. **</releases>**
9. **<snapshots>**
10. **<enabled>**true**</enabled>**
11. **</snapshots>**
12. **</repository>**
13. **</repositories>**

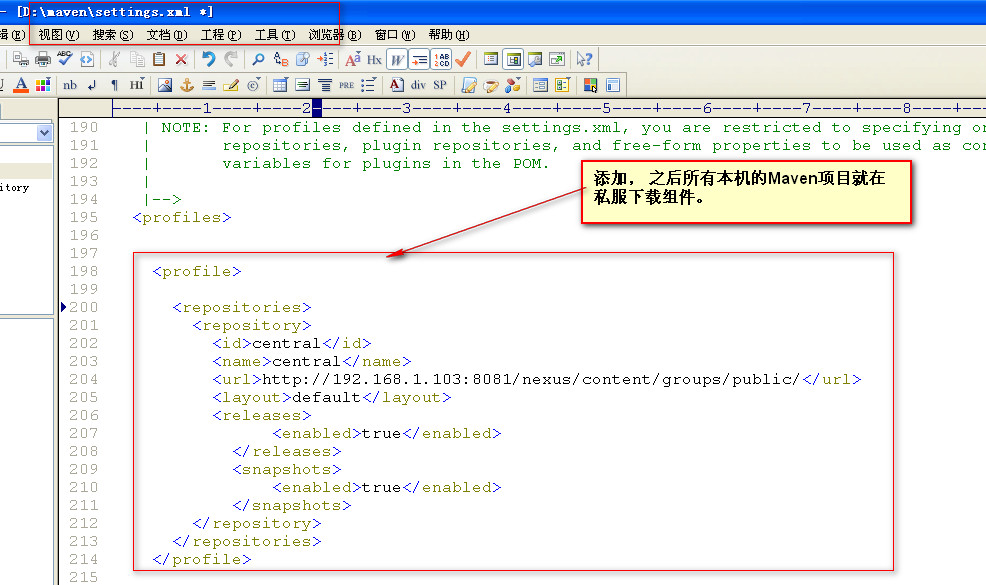
**指定插件仓库**

**[html]** [view plaincopy](http://blog.csdn.net/shenshen123jun/article/details/9084293)

1. **<pluginRepositories>**
2. **<pluginRepository>**
3. **<id>**nexus**</id>**
4. **<name>**nexus**</name>**
5. **<url>**http://192.168.1.103:8081/nexus/content/groups/public/**</url>**
6. **<releases>**
7. **<enabled>**true**</enabled>**
8. **</releases>**
9. **<snapshots>**
10. **<enabled>**true**</enabled>**
11. **</snapshots>**
12. **</pluginRepository>**
13. **</pluginRepositories>**

**这样只有本项目才在私服下载组件  
这样这个Maven项目构建的时候会从私服下载相关依赖。当然这个配置仅仅是在此项目中生效，对于其他项目还是不起作用。如果相对Maven的其他项目也生效的话。需要修改全局的settings.xml文件。**

**修改settings.xml为**



**追加激活profile**

**[html]** [view plaincopy](http://blog.csdn.net/shenshen123jun/article/details/9084293)

1. **<activeProfiles>**
2. **<activeProfile>**central**</activeProfile>**
4. **</activeProfiles>**

**之后所有本机的Maven项目就在私服下载组件。（这样比较好）**

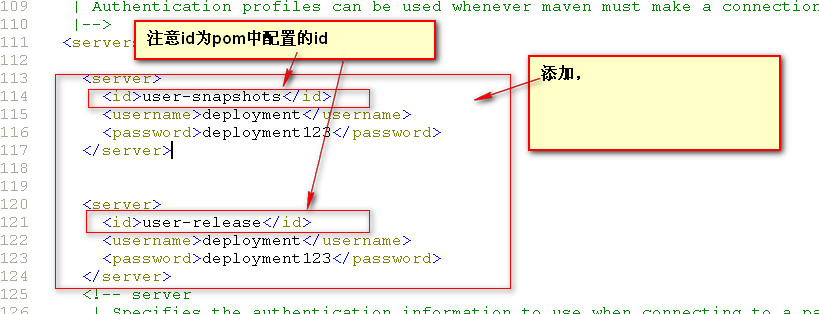
**项目的发布**

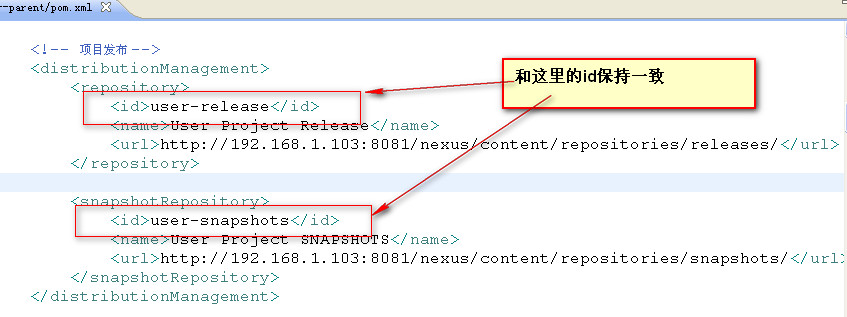
**[html]** [view plaincopy](http://blog.csdn.net/shenshen123jun/article/details/9084293)

1. **<distributionManagement>**
2. **<repository>**
3. **<id>**user-release**</id>**
4. **<name>**User Project Release**</name>**
5. **<url>**http://192.168.1.103:8081/nexus/content/repositories/releases/**</url>**
6. **</repository>**
8. **<snapshotRepository>**
9. **<id>**user-snapshots**</id>**
10. **<name>**User Project SNAPSHOTS**</name>**
11. **<url>**http://192.168.1.103:8081/nexus/content/repositories/snapshots/**</url>**
12. **</snapshotRepository>**
13. **</distributionManagement>**

**注意配置了还是发布项目到私服失败，原因为没有权限，会出现401错误码，原因就是权限不够。**

**配置权限在settings.xml**





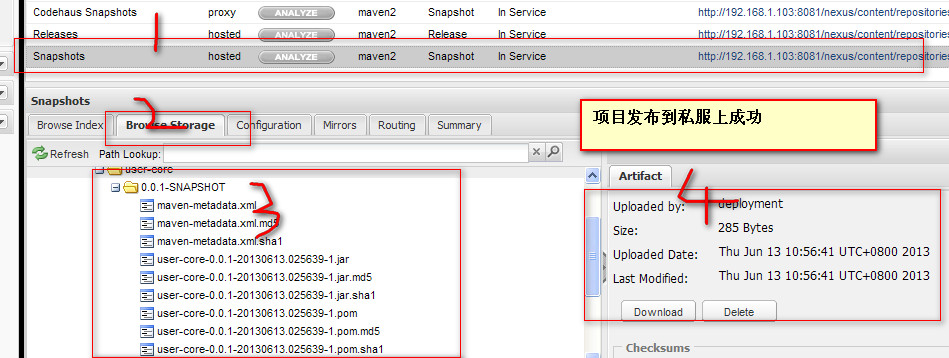
**注意Respository中的id一定要和server下的id一致，切记！！否则出现权限问题。**

**然后运行发布**

**clean deploy**

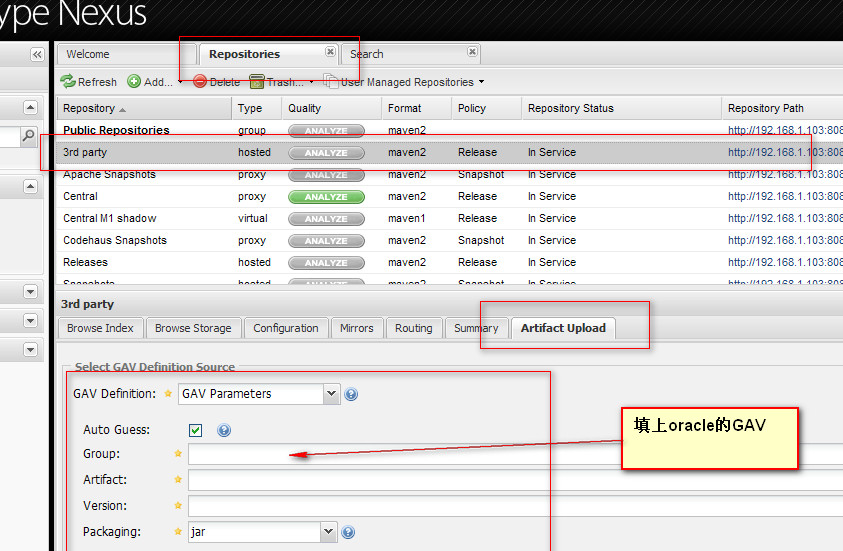
**在控制台发布成功**

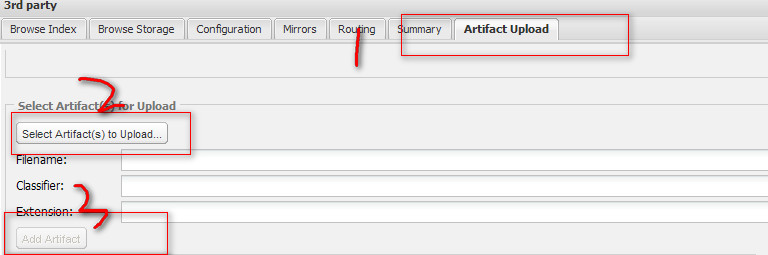
**然后进入到私服上的仓库中，看一下确实存在刚刚发布的项目**



**宿主库——3rd party**

**假如我们下载了Oracle的驱动程序jar包想给其他项目组使用，就需要上传该jar包。选中宿主库——3rd party，之后选择Artifact Upload上传至宿主空间。**





最后点击上传

