

要停止Nexus，Ctrl+C即可，此外InstallNexus.bat可以用来将Nexus安装成一个windows服务，其余的脚本则对应了启动，停止，暂停，恢复，卸载Nexus服务。

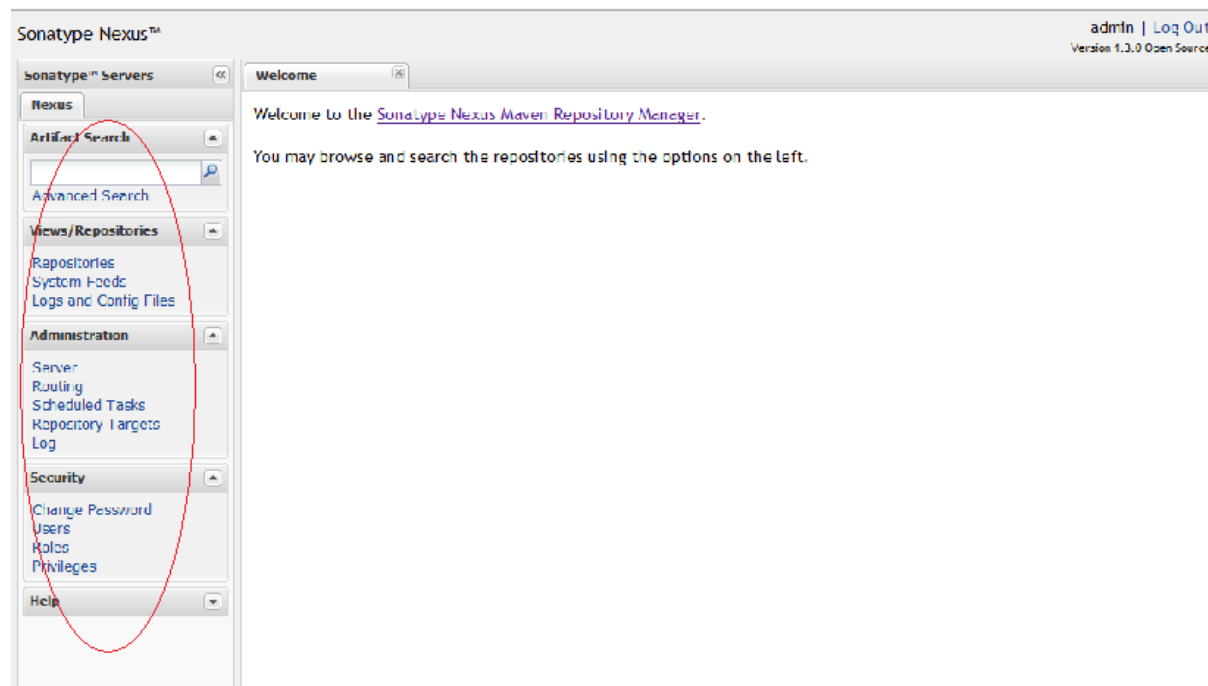
### ***WAR方式安装***

你需要有一个能运行的web容器，这里以Tomcat为例，加入Tomcat的安装目录位于 *D:\dev\_tools\apache-tomcat-6.0.18*，首先我们将下载的 *nexus-webapp-1.3.0.war* 重命名为 *nexus.war*，然后复制到 *D:\dev\_tools\apache-tomcat-6.0.18\webapps\nexus.war*，然后启动CMD，cd到 *D:\dev\_tools\apache-tomcat-6.0.18\bin\* 目录，运行**startup.bat**。一切OK，现在可以打开浏览器访问 <http://127.0.0.1:8080/nexus>，你会得到和上图一样的界面。

## 代理外部Maven仓库

### ***登陆***

要管理Nexus，你首先需要以管理员身份登陆，点击界面右上角的login，输入默认的登录名和密码：admin/admin123，登陆成功后，你会看到左边的导航栏增加了很多内容：



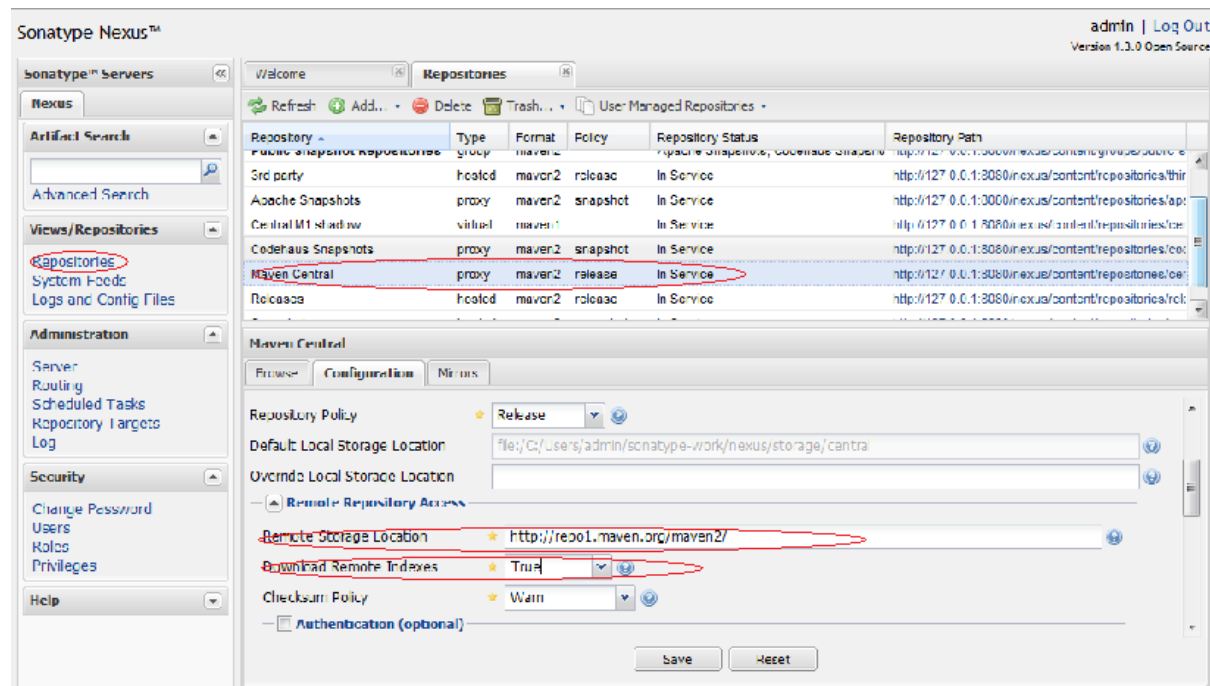
这里，可以管理仓库，配置Nexus系统，管理任务，管理用户，角色，权限，查看系统的RSS源，管理及查看系统日志，等等。你会看到Nexus的功能十分丰富和强大，本文，笔者只介绍一些最基本的管理和操作。

## 代理Maven中央仓库

点击左边导航栏的Repositories，界面的主面板会显示所有仓库及仓库组的列表，你会看到它们的Type字段的值有group，hosted，proxy，virtual。这里我们不关心virtual，只介绍下另外三种类型：

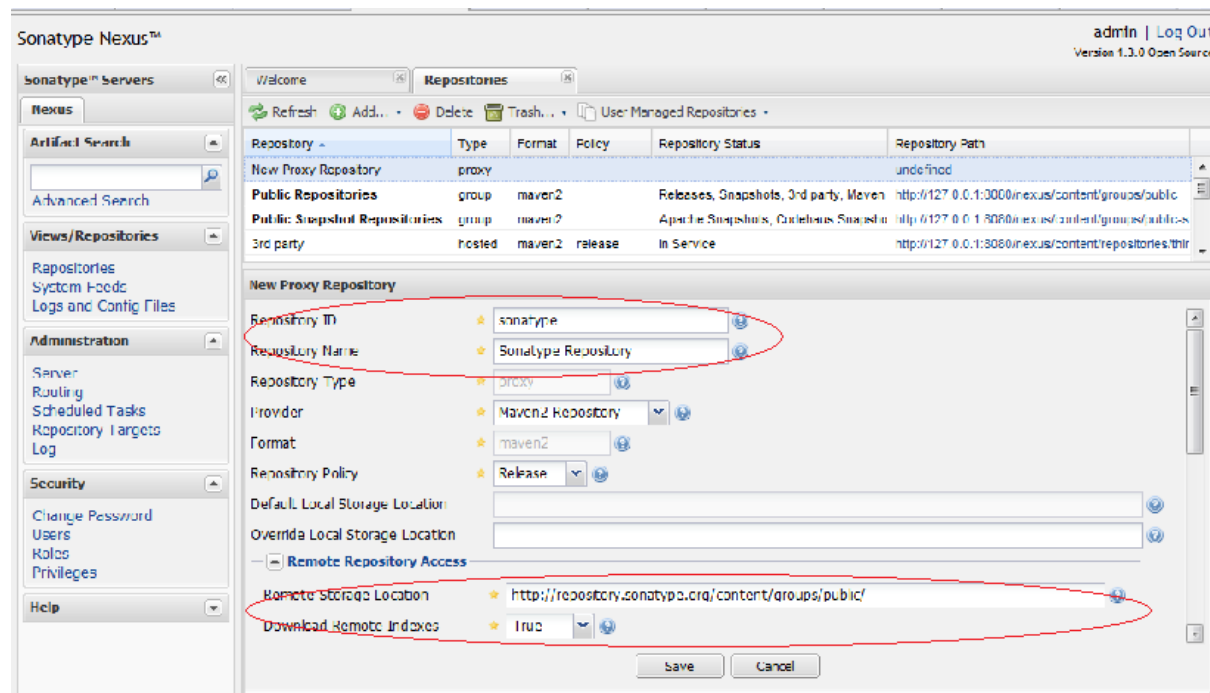
- hosted，本地仓库，通常会部署自己的构件到这一类型的仓库。
- proxy，代理仓库，它们被用来代理远程的公共仓库，如maven中央仓库。
- group，仓库组，用来合并多个hosted/proxy仓库，通常我们配置maven依赖仓库组。

由此我们知道，我们需要配置一个Maven中央仓库的proxy，其实Nexus已经内置了Maven Central，但我们需要做一些配置。点击仓库列表中的Maven Central，你会注意到它的Policy是release，这说明它不会代理远程仓库的snapshot构件，这是有原因的，远程仓库的snapshot版本构件不稳定且不受你控制，使用这样的构件含有潜在的风险。然后我们发现主面板下方有三个Tab，分别为Browse，Configuration和Mirrors，我们点击Configuration进行配置，你现在需要关心的是两个配置项：“Remote Storage Location”为远程仓库的地址，对于Maven Central来说是http://repo1.maven.org/maven2/；“Download Remote Indexes”顾名思义是指是否下载远程索引文件，Maven Central的该字段默认为False，这是为了防止大量Nexus无意识的去消耗中央仓库的带宽（中央仓库有大量的构件，其索引文件也很大）。这里我们需要将其设置为True，然后点击Save。在Nexus下载的中央仓库索引文件之后，我们就可以在本地搜索中央仓库的所有构件。下图展示了我们刚才所涉及的配置：



## 添加一个代理仓库

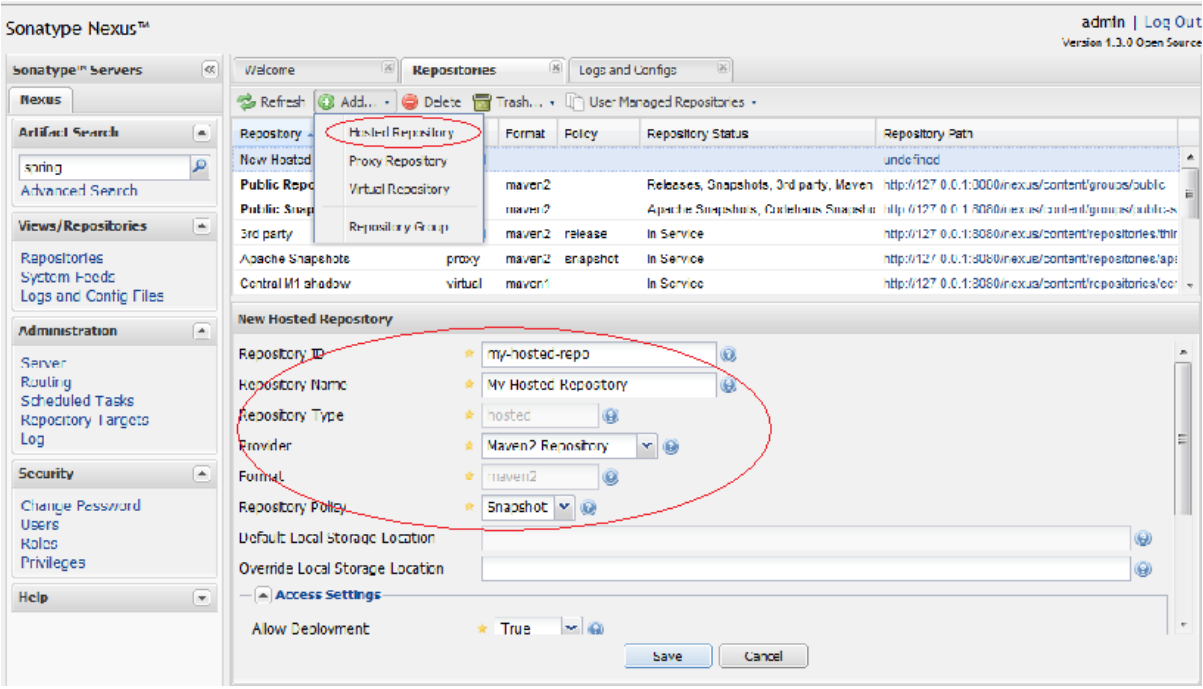
这里我们再举一个例子，我们想要代理Sonatype的公共仓库，其地址为：<http://repository.sonatype.org/content/groups/public/>。步骤如下，在Repositories面板的上方，点击Add，然后选择Proxy Repository，在下方的配置部分，我们填写如下的信息：Repository ID - sonatype；Repository Name - Sonatype Repository；Remote Storage Location - <http://repository.sonatype.org/content/groups/public/>。其余的保持默认值，需要注意的是Repository Policy，我们不想代理snapshot构件，原因前面已经描述。然后点击Save。配置页面如下：



## 管理本地Maven仓库

Nexus预定义了3个本地仓库，分别为Releases，Snapshots，和3rd Party。这三个仓库都有各自明确的目的。Releases用于部署我们自己的release构件，Snapshots用于部署我们自己的snapshot构件，而3rd Party用于部署第三方构件，有些构件如Oracle的JDBC驱动，我们不能从公共仓库下载到，我们就需要将其部署到自己的仓库中。

当然你也可以创建自己的本地仓库，步骤和创建代理仓库类似，点击Repository面板上方的Add按钮，然后选择Hosted Repository，然后在下方的配置面板中输入id和name，注意这里我们不再需要填写远程仓库地址，Repository Type则为不可修改的hosted，而关于Repository Policy，你可以根据自己的需要选择Release或者Snapshot，如图：

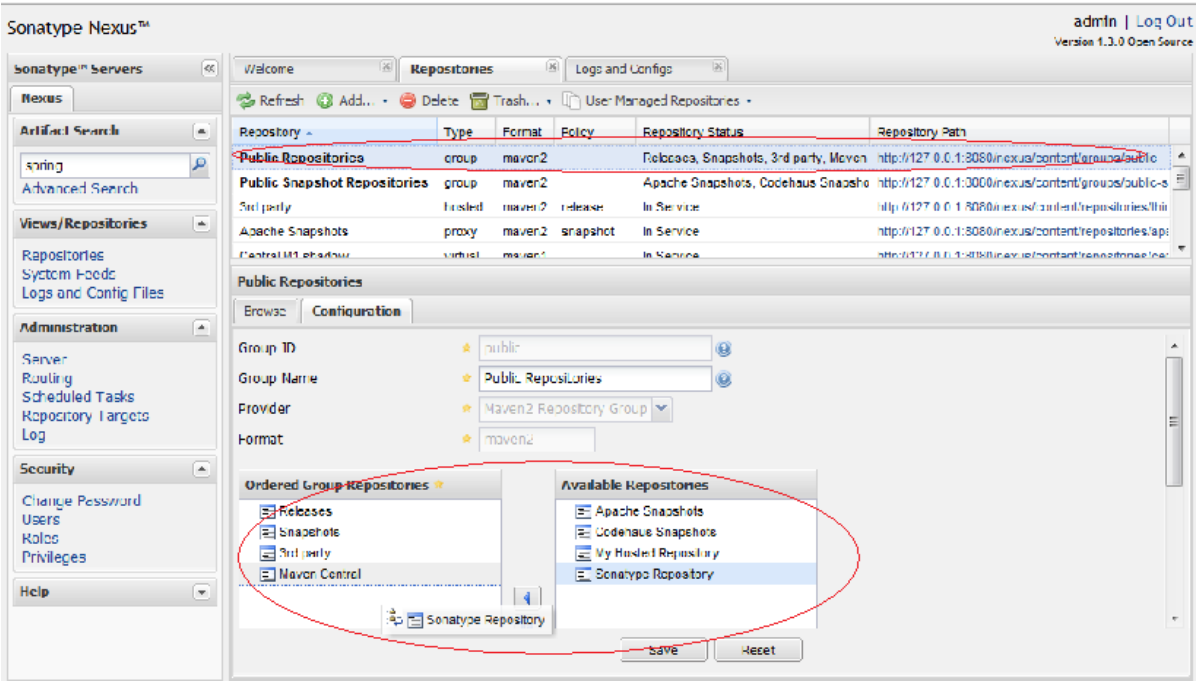


## 管理Maven仓库组

Nexus中仓库组的概念是Maven没有的，在Maven看来，不管你是hosted也好，proxy也好，或者group也好，对我都是一样的，我只管根据groupid，artifactId，version等信息向你要构件。为了方便Maven的配置，Nexus能够将多个仓库，hosted或者proxy合并成一个group，这样，Maven只需要依赖于一个group，便能使用所有该group包含的仓库的内容。

Nexus预定义了“Public Repositories”和“Public Snapshot Repositories”两个仓库组，前者默认合并所有预定义的Release仓库，后者默认合并所有预定义的Snapshot仓库。我们在本文前面的部分创建了一个名为“Sonatype Repository”的仓库，现在将其合并到“Public Repositories”中。

点击仓库列表中的“Public Repositories”，然后选择下方的"Configuration" Tab，在配置面板中，将右边“Available Repositories”中的“Sonatype Repository”拖拽到左边的“Ordered Group Repositories”中，如图：

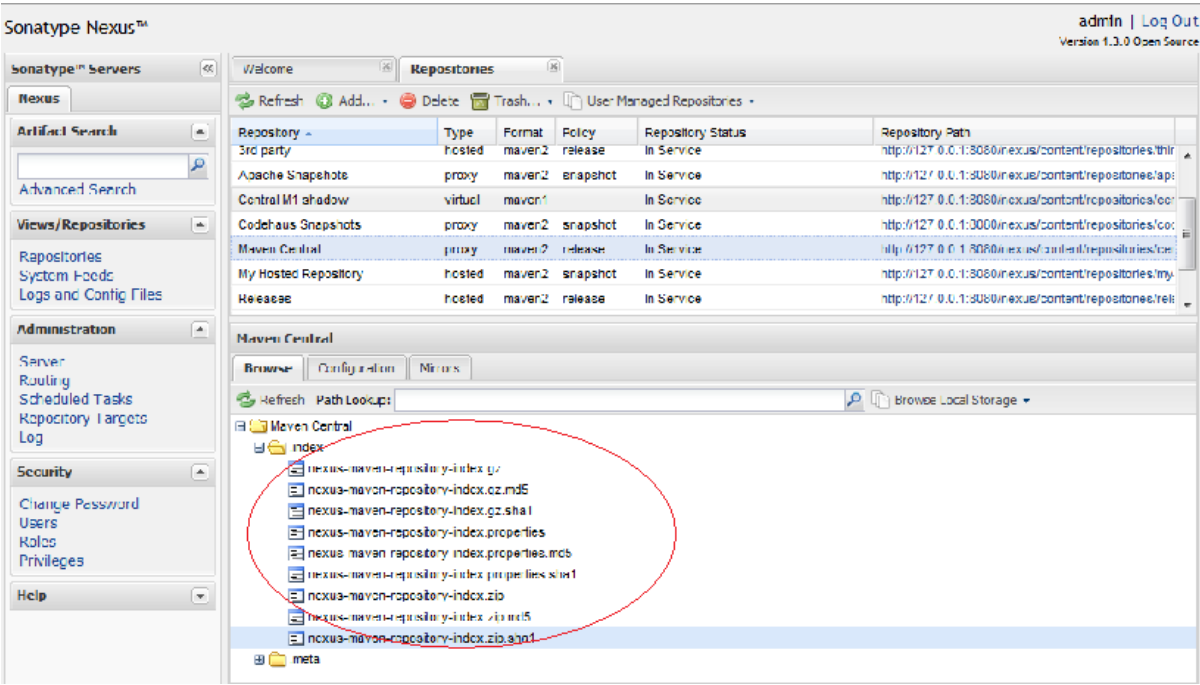




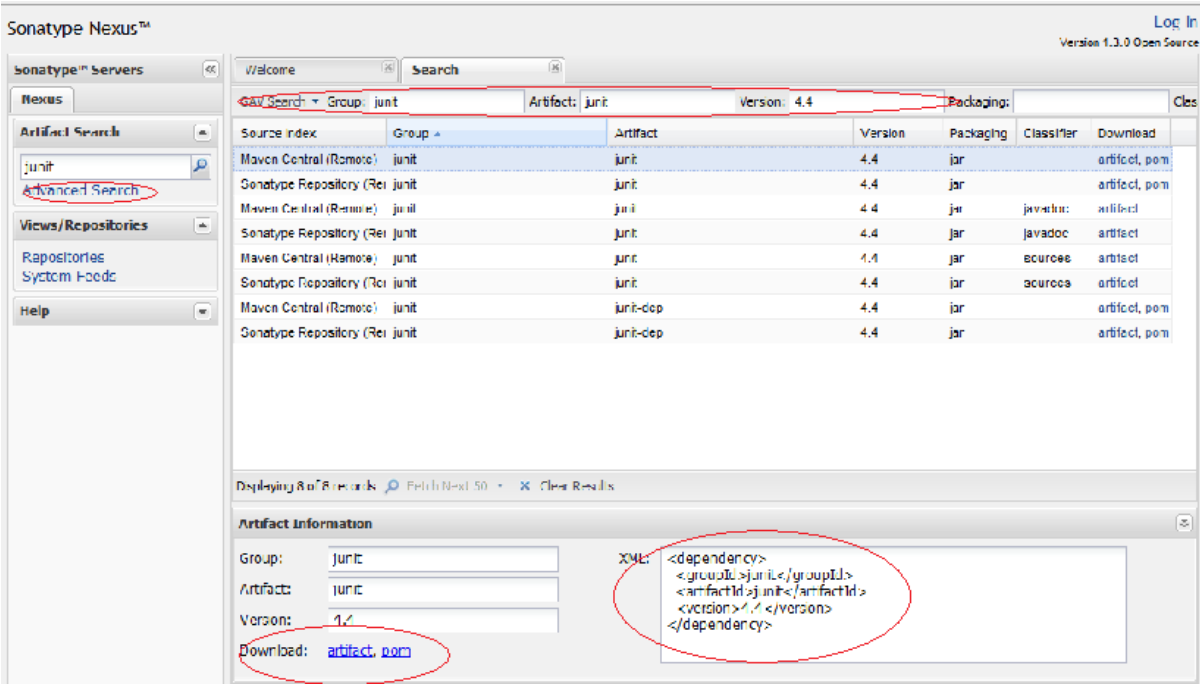
创建仓库组和创建proxy及hosted仓库类似，这里不再赘述。需要注意的是format字段需要填写 “maven2”，添加你感兴趣的仓库即可。

## 搜索构件

在浩大的Maven仓库中一下下点击链接，浏览路径以寻找感兴趣的构件是一件很郁闷的事情。Nexus基于nexus-indexer提供构件搜索功能，要想对仓库进行搜索，无论是hosted，proxy，或者group，你都必须确认索引文件存在。这一点对于代理仓库尤其重要，有些远程仓库可能根本就没有索引，所以你无法搜索这些代理仓库。有些远程仓库的远程索引非常大，如中央仓库达到了70M左右，那么第一次下载索引需要花很多时间，所以要期望得到搜索结果，确保看到如下的文件：



一旦你的Nexus拥有了本地或者远程仓库的索引文件，你就可以享受Nexus的构件搜索功能了。不论登陆与否，你都可以使用关键字进行模糊搜索，比如我在左边导航栏上部的搜索框内输入junit，然后点击搜索按钮，右边立刻会分页显示500多条的junit相关构件信息。如果你了解更多的信息，你也可以通过限定groupId，artifactId，version进行搜索，点击导航栏中的“Advanced Search”，点击右边所有页面左上角的下拉框，选择“GAV Search”。笔者这里输入junit:junit:4.4，然后回车：




选择一项搜索结果，在页面下方会显示“Artifact Information”的面板，你可以点击"artifact"或者"pom"下载对应文件，而该面板右边更显示了一个Maven依赖配置，你可以直接复制该配置到Maven POM中，这是个十分方便的特性。

此外，值得一提的是，Nexus还支持基于classname的搜索，你只需点击搜索页面右上角的下拉框，选择 “Classname Search” ，然后输入类名即可，这里我不再赘述。

## 配置Maven使用Nexus

默认情况下，Maven依赖于中央仓库，这是为了能让Maven开箱即用，但仅仅这么做明显是错误的，这会造成大量的时间及带宽的浪费。既然文章的前面已经介绍了如何安装和配置Nexus，现在我们就要配置Maven来使用本地的Nexus，以节省时间和带宽资源。

我们可以将Repository配置到POM中，但一般来说这不是很好的做法，原因很简单，你需要为所有的Maven项目重复该配置。因此，这里我将Repository的配置放到\$user\_home/.m2/settings.xml中：

Xml代码 

```
1. <settings>
2. ...
3. <profiles>
4.   <profile>
5.     <id>dev</id>
6.     <repositories>
7.       <repository>
8.         <id>local-nexus</id>
9.         <url>http://127.0.0.1:8080/nexus/content/groups/public/</url>
10.        <releases>
11.          <enabled>true</enabled>
12.        </releases>
13.        <snapshots>
14.          <enabled>true</enabled>
15.        </snapshots>
16.      </repository>
17.    </repositories>
18.  </profile>
19. </profiles>
20. <activeProfiles>
21.   <activeProfile>dev</activeProfile>
22. </activeProfiles>
23. ...
24. </settings>
```

由于我们不能直接在settings.xml中插入<repositories>元素，这里我们编写了一个profile，并添加了一个profile并使用<activeProfile>元素自动将这个profile激活。这里的local-nexus仓库指向了刚才我们配置的Nexus中 “Public Repositories” 仓库组，也就是说，所有该仓库组包含的仓库都能供我们使用。此外，我们通过<releases>和<snapshots>元素激活了Maven对于仓库所有类型构件下载的支持，当然你也可以调节该配置，比如说禁止Maven从Nexus下载snapshot构件。

使用该配置，Maven就会从你的Nexus服务器下载构件了，速度和从Central下载可不是一个数量级的。

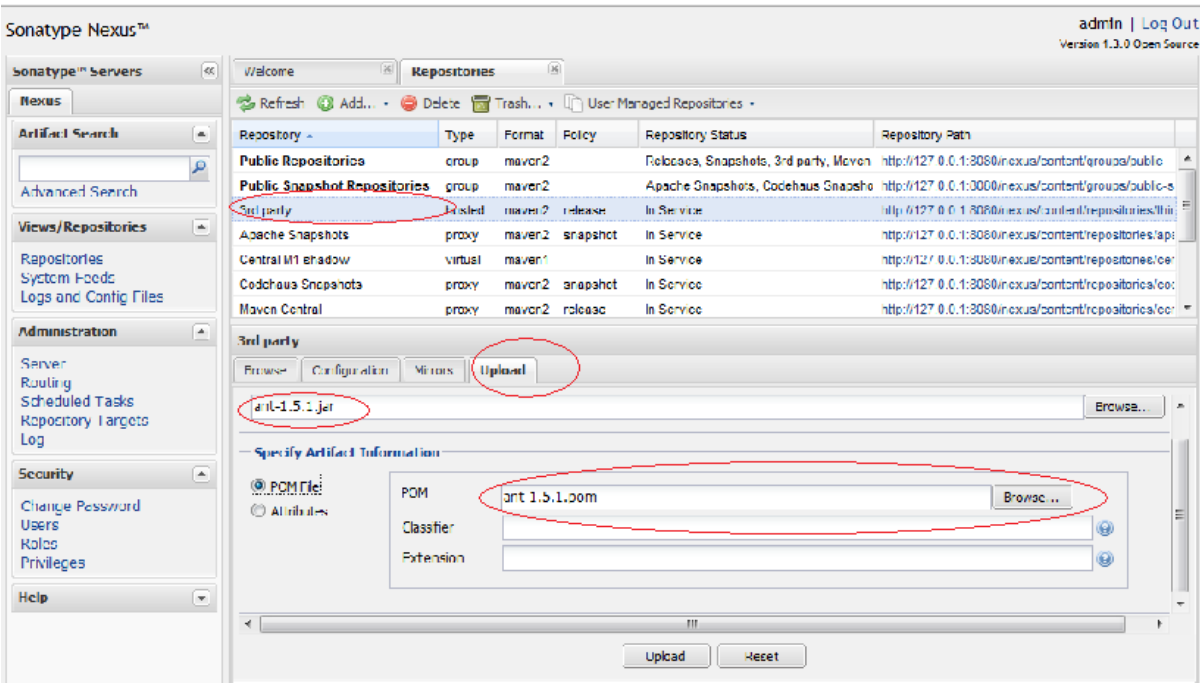
## 部署构件至Nexus

Nexus提供了两种方式部署构件，你可以从UI直接上传，也可以配置Maven部署构件。

通过Nexus UI部署

有时候有个jar文件你无法从公共Maven仓库找到，但是你能从其它得到这个jar文件（甚至是POM），那么你完全可以将这个文件部署到Nexus中，使其成为标准流程的一部分。步骤如下：

点击左边导航栏的"Repository"，在右边的仓库列表选择一个仓库，如“3rd Party”，然后会看到页面下方有四个tab，选择最后一个“Upload”，你会看到构件上传界面。选择你要上传的构件，并指定POM，（或者手工编写GAV等信息），最后点击Upload，该构件就直接被部署到了Nexus的"3rd Party"仓库中。如图：



通过Maven部署

更常见的用例是：团队在开发一个项目的各个模块，为了让自己开发的模块能够快速让其他人使用，你会想要将snapshot版本的构件部署到Maven仓库中，其他人只需要在POM添加一个对于你开发模块的依赖，就能随时拿到最新的snapshot。

以下的pom.xml配置和settings.xml能让你通过Maven自动化部署构件：

pom.xml

Xml代码 ☆

1. **<project>**
2. ...
3. **<distributionManagement>**
4. **<repository>**
5. **<id>nexus-releases</id>**
6. **<name>Nexus Release Repository</name>**
7. **<url>http://127.0.0.1:8080/nexus/content/repositories/releases/</url>**
8. **</repository>**
9. **<snapshotRepository>**
10. **<id>nexus-snapshots</id>**
11. **<name>Nexus Snapshot Repository</name>**
12. **<url>http://127.0.0.1:8080/nexus/content/repositories/snapshots/</url>**



```
13. </snapshotRepository>
14. </distributionManagement>
15. ...
16. </project>
```

settings.xml

Xml代码 ☆

```
1. <settings>
2. ...
3. <servers>
4.   <server>
5.     <id>nexus-releases</id>
6.     <username>admin</username>
7.     <password>admin123</password>
8.   </server>
9.   <server>
10.    <id>nexus-snapshots</id>
11.    <username>admin</username>
12.    <password>admin123</password>
13.  </server>
14. </servers>
15. ...
16. </settings>
```

这里我们配置所有的snapshot版本构件部署到Nexus的Snapshots仓库中，所有的release构件部署到Nexus的Releases仓库中。由于部署需要登陆，因为我们在settings.xml中配置对应Repository id的用户名和密码。

然后，在项目目录中执行**mvn deploy**，你会看到maven将项目构件部署到Nexus中，浏览Nexus对应的仓库，就可以看到刚才部署的构件。当其他人构建其项目时，Maven就会从Nexus寻找依赖并下载。

## 总结

本文介绍强大的仓库管理器——Nexus，包括如何下载安装Nexus，配置Nexus代理中央仓库，管理Nexus的代理仓库，本地仓库，以及仓库组。并帮助你了解如何通过Nexus搜索构件。最后，如何在Maven中配置Nexus仓库，以及如何部署构件到Nexus仓库中。这些都是Nexus中最基本也是最常用的功能。随着使用的深入，你会发现Nexus还有很多其它的特性，如用户管理，角色权限管理等等。

Nexus的OSS版本是完全开源的，如果你有兴趣，你可以学习其源码，甚至自己实现一个REST客户端。

马上拥抱Nexus吧，它是免费的！



- 2009-03-15 23:07
- 浏览 126343
- [评论\(43\)](#)
- [查看更多](#)

相关资源推荐

[3分钟带你了解iPhone7升级iOS11所有问题](#)

[Python Crash Course](#)

[微信小程序 VS 原生App](#)

[教你怎么免费搭建discuz论坛教程](#)

[Wi-Fi 爆重大安全漏洞，Android、iOS、Windows 等所有无线设备都不安全了](#)

[Nexus入门指南（图文）](#)

[Nexus入门指南\(图文\).docx](#)

[思科Nexus\\_7000入门经典](#)

[nexus 2.x 安装配置指南](#)

[思科Nexus\\_7000入门经典](#)

[Java数据结构与算法解析\(一\)——表](#)

[关注CSDN程序人生公众号，轻松获得下载积分](#)

[操作系统OEM DIY工具](#)

[一小时学会搭建网站](#)

[黑客基础知识大全（TXT）](#)

[Nexus入门指南（图文）](#)

[Nexus1.1中文版指南](#)

[Nexus\\_7000入门经典\\_N7K\\_VDC\\_VPC](#)

[Cisco Nexus 1000V安装指南](#)

[OpenCMS内容管理入门指南（300页中文）.pdf](#)

参考知识库



[Java SE知识库](#) 29023 关注 / 578 收录



[Java 知识库](#) 36111 关注 / 3748 收录



[Java EE知识库](#) 23315 关注 / 1416 收录



[JavaScript知识库](#) 17271 关注 / 1517 收录



[jQuery知识库](#) 10581 关注 / 948 收录



[AngularJS知识库](#) 5503 关注 / 590 收录

## 评论

43 楼 [Wuaner](#) 2014-11-26

使用 nexus 时，除了在 ~/.m2/settings.xml 中通过 mirrors / profiles 外，有没有更好的方式来覆盖 super POM 中的 central repo ?

42 楼 [wangshu3000](#) 2014-04-24

为什么我在3rd Party库里上传个jar会在Release库里出现，3rd Party库还是空的。😞

41 楼 [houzhanshanlinzhou](#) 2014-04-22

写得非常不错！

40 楼 [hanmiao](#) 2013-08-27

xisuchi 写道

貌似博主已经不在ITeye了....😞😞

博主自己新开了独立域名的博客，在这里：<http://www.juvenxu.com/>

39 楼 [xisuchi](#) 2013-07-18

貌似博主已经不在ITeye了....😞😞

38 楼 [jjava33](#) 2013-05-16

麻烦问一下，我按照 通过Nexus UI部署第三方构件，在nexus中已经成功上传，界面中能看到第三方构件，可是，在开发工具eclipse中报错 missing artifact ，查看本地库，发现没有成功下载，不知是什么原因，请您方便的话，能予以答复，谢谢。

37 楼 [liuqing9382](#) 2012-06-28

图挂了，楼主能补上么？

36 楼 [niuqiang2008](#) 2012-03-29

谢谢您的回答 ... 可惜我不在IT界混了。

35 楼 [goodboylll](#) 2012-03-24

你好，我和6楼的朋友遇见了同样的问题

当我update、repair index的时候 log也是报错

jvm 1 | java.io.FileNotFoundException: nexus-maven-repository-index.properties (item not found) 总是找不到这个文件

但是我看nexus里面的路径指向 是正确的

而且相应的文件也是存在的 就是报错无法找打配置文件

求教啊 谢谢了

34 楼 [yangpeihai](#) 2011-12-28

谢谢，我是在fedora下跑多，非常顺利，收藏啦😊

33 楼 [wmj007](#) 2011-12-16

探索中，学习了

32 楼 [maskwerewolf](#) 2011-09-29

org.apache.maven.mercury.repository.metadata.MetadataException

mvn deploy nexus 报错，求解

31 楼 [C.T](#) 2011-09-28

请问用Nexus可以直接下载构件到本地吗？我在Nexus里可以浏览中央仓库的东西，不过不知道怎样下载到本地啊！

30 楼 [juvenshun](#) 2011-01-08

niuqiang2008 写道

希望您能帮我看看 ... 一直没解决

换jdk6或者较新版本的jdk5试试吧

29 楼 [niuqiang2008](#) 2011-01-08

希望您能帮我看看 ... 一直没解决

28 楼 [niuqiang2008](#) 2011-01-08

你好 ,我用如上两种方法都启动不了nexus,报错如下:

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
javax.servlet.ServletException: Servlet.init() for servlet nexus threw exception
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

root cause

```
java.lang.IllegalStateException: The PlexusServerServlet couldn't lookup the target component (role='org.restlet.Application', hint='nexus')
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:58)
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)
javax.servlet.GenericServlet.init(GenericServlet.java:212)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

root cause

org.codehaus.plexus.component.repository.exception.ComponentLookupException: com.google.inject.ProvisionException: Guice provision errors:

```
1) Error injecting: org.sonatype.nexus.rest.NexusApplication
  at ClassRealm[plexus.core, parent: null]
  at ClassRealm[plexus.core, parent: null]
  while locating org.restlet.Application annotated with @com.google.inject.name.Named(value=nexus)
```

1 error

```
  role: org.restlet.Application
  roleHint: nexus
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:238)
```

```
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:214)
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:49)
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)
javax.servlet.GenericServlet.init(GenericServlet.java:212)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

root cause

com.google.inject.ProvisionException: Guice provision errors:

```
1) Error injecting: org.sonatype.nexus.rest.NexusApplication
  at ClassRealm[plexus.core, parent: null]
  at ClassRealm[plexus.core, parent: null]
  while locating org.restlet.Application annotated with @com.google.inject.name.Named(value=nexus)
```

1 error

```
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:952)
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:234)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:214)
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:49)
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)
javax.servlet.GenericServlet.init(GenericServlet.java:212)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

root cause

com.google.inject.ProvisionException: Guice provision errors:

```
1) Error injecting: org.sonatype.nexus.rest.status.StatusPlexusResource
  at ClassRealm[plexus.core, parent: null]
  at ClassRealm[plexus.core, parent: null]
  while locating org.sonatype.plexus.rest.resource.ManagedPlexusResource annotated with @com.google.inject.name.Named(value=StatusPlexusResource)
  while locating org.sonatype.nexus.rest.NexusApplication
```



1 error

```
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:952)
com.google.inject.internal.InjectorImpl.getInstance(InjectorImpl.java:978)
org.sonatype.guice.bean.reflect.AbstractDeferredClass.get(AbstractDeferredClass.java:48)
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)
com.google.inject.internal.ProviderToInternalFactoryAdapter$1.call(ProviderToInternalFactoryAdapter.java:46)
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)
com.google.inject.internal.ProviderToInternalFactoryAdapter.get(ProviderToInternalFactoryAdapter.java:40)
com.google.inject.Scopes$1$1.get(Scopes.java:63)
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)
com.google.inject.internal.InjectorImpl$4$1.call(InjectorImpl.java:943)
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:989)
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:939)
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:234)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:214)
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:49)
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)
javax.servlet.GenericServlet.init(GenericServlet.java:212)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

root cause

```
java.lang.TypeNotPresentException: Type org.codehaus.enunciate.contract.jaxrs.ResourceMethodSignature not present
sun.reflect.generics.factory.CoreReflectionFactory.makeNamedType(CoreReflectionFactory.java:98)
sun.reflect.generics.visitor.Reifier.visitClassTypeSignature(Reifier.java:107)
sun.reflect.generics.tree.ClassTypeSignature.accept(ClassTypeSignature.java:31)
sun.reflect.annotation.AnnotationParser.parseSig(AnnotationParser.java:351)
sun.reflect.annotation.AnnotationParser.parseAnnotation(AnnotationParser.java:175)
sun.reflect.annotation.AnnotationParser.parseAnnotations2(AnnotationParser.java:69)
sun.reflect.annotation.AnnotationParser.parseAnnotations(AnnotationParser.java:52)
java.lang.reflect.Method.getDeclaredAnnotations(Method.java:676)
java.lang.reflect.Method.getAnnotation(Method.java:663)
com.google.inject.spi.InjectionPoint.getAtInject(InjectionPoint.java:466)
com.google.inject.spi.InjectionPoint.getInjectionPoints(InjectionPoint.java:664)
com.google.inject.spi.InjectionPoint.forInstanceMethodsAndFields(InjectionPoint.java:356)
com.google.inject.internal.ConstructorBindingImpl.getInternalDependencies(ConstructorBindingImpl.java:151)
com.google.inject.internal.InjectorImpl.getInternalDependencies(InjectorImpl.java:559)
com.google.inject.internal.InjectorImpl.cleanup(InjectorImpl.java:517)
com.google.inject.internal.InjectorImpl.initializeJitBinding(InjectorImpl.java:503)
```

com.google.inject.internal.InjectorImpl.createJustInTimeBinding(InjectorImpl.java:812)  
com.google.inject.internal.InjectorImpl.createJustInTimeBindingRecursive(InjectorImpl.java:744)  
com.google.inject.internal.InjectorImpl.getJustInTimeBinding(InjectorImpl.java:233)  
com.google.inject.internal.InjectorImpl.getBindingOrThrow(InjectorImpl.java:184)  
com.google.inject.internal.InjectorImpl.getInternalFactory(InjectorImpl.java:818)  
com.google.inject.internal.InjectorImpl.getProviderOrThrow(InjectorImpl.java:932)  
com.google.inject.internal.InjectorImpl.getProvider(InjectorImpl.java:965)  
com.google.inject.internal.InjectorImpl.getProvider(InjectorImpl.java:926)  
com.google.inject.internal.InjectorImpl.getInstance(InjectorImpl.java:978)  
org.sonatype.guice.bean.reflect.AbstractDeferredClass.get(AbstractDeferredClass.java:48)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.ProviderToInternalFactoryAdapter\$1.call(ProviderToInternalFactoryAdapter.java:46)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)  
com.google.inject.internal.ProviderToInternalFactoryAdapter.get(ProviderToInternalFactoryAdapter.java:40)  
com.google.inject.Scopes\$1\$1.get(Scopes.java:63)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.InjectorImpl\$4\$1.call(InjectorImpl.java:943)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)  
com.google.inject.internal.InjectorImpl\$4.get(InjectorImpl.java:939)  
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)  
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)  
org.sonatype.guice.plexus.binders.PlexusRequirements\$RequirementProvider.get(PlexusRequirements.java:222)  
org.sonatype.guice.plexus.binders.ProvidedPropertyBinding.injectProperty(ProvidedPropertyBinding.java:50)  
org.sonatype.guice.bean.inject.BeanInjector.injectMembers(BeanInjector.java:53)  
com.google.inject.internal.MembersInjectorImpl.injectMembers(MembersInjectorImpl.java:120)  
com.google.inject.internal.ConstructorInjector.construct(ConstructorInjector.java:94)  
com.google.inject.internal.ConstructorBindingImpl\$Factory.get(ConstructorBindingImpl.java:253)  
com.google.inject.internal.InjectorImpl\$4\$1.call(InjectorImpl.java:943)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)  
com.google.inject.internal.InjectorImpl\$4.get(InjectorImpl.java:939)  
com.google.inject.internal.InjectorImpl.getInstance(InjectorImpl.java:978)  
org.sonatype.guice.bean.reflect.AbstractDeferredClass.get(AbstractDeferredClass.java:48)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.ProviderToInternalFactoryAdapter\$1.call(ProviderToInternalFactoryAdapter.java:46)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)  
com.google.inject.internal.ProviderToInternalFactoryAdapter.get(ProviderToInternalFactoryAdapter.java:40)  
com.google.inject.Scopes\$1\$1.get(Scopes.java:63)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.InjectorImpl\$4\$1.call(InjectorImpl.java:943)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:989)  
com.google.inject.internal.InjectorImpl\$4.get(InjectorImpl.java:939)  
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)  
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)  
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:234)  
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:214)  
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:49)  
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)  
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)  
javax.servlet.GenericServlet.init(GenericServlet.java:212)

org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)  
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)  
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)  
org.apache.coyote.http11.Http11Protocol\$Http11ConnectionHandler.process(Http11Protocol.java:588)  
org.apache.tomcat.util.net.JIoEndpoint\$Worker.run(JIoEndpoint.java:489)  
java.lang.Thread.run(Thread.java:595)

root cause

java.lang.ClassNotFoundException: org.codehaus.enunciate.contract.jaxrs.ResourceMethodSignature  
org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1516)  
org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1361)  
java.lang.ClassLoader.loadClassInternal(ClassLoader.java:319)  
java.lang.Class.forName0(Native Method)  
java.lang.Class.forName(Class.java:242)  
sun.reflect.generics.factory.CoreReflectionFactory.makeNamedType(CoreReflectionFactory.java:95)  
sun.reflect.generics.visitor.Reifier.visitClassTypeSignature(Reifier.java:107)  
sun.reflect.generics.tree.ClassTypeSignature.accept(ClassTypeSignature.java:31)  
sun.reflect.annotation.AnnotationParser.parseSig(AnnotationParser.java:351)  
sun.reflect.annotation.AnnotationParser.parseAnnotation(AnnotationParser.java:175)  
sun.reflect.annotation.AnnotationParser.parseAnnotations2(AnnotationParser.java:69)  
sun.reflect.annotation.AnnotationParser.parseAnnotations(AnnotationParser.java:52)  
java.lang.reflect.Method.getDeclaredAnnotations(Method.java:676)  
java.lang.reflect.Method.getAnnotation(Method.java:663)  
com.google.inject.spi.InjectionPoint.getAtInject(InjectionPoint.java:466)  
com.google.inject.spi.InjectionPoint.getInjectionPoints(InjectionPoint.java:664)  
com.google.inject.spi.InjectionPoint.forInstanceMethodsAndFields(InjectionPoint.java:356)  
com.google.inject.internal.ConstructorBindingImpl.getInternalDependencies(ConstructorBindingImpl.java:151)  
com.google.inject.internal.InjectorImpl.getInternalDependencies(InjectorImpl.java:559)  
com.google.inject.internal.InjectorImpl.cleanup(InjectorImpl.java:517)  
com.google.inject.internal.InjectorImpl.initializeJitBinding(InjectorImpl.java:503)  
com.google.inject.internal.InjectorImpl.createJustInTimeBinding(InjectorImpl.java:812)  
com.google.inject.internal.InjectorImpl.createJustInTimeBindingRecursive(InjectorImpl.java:744)  
com.google.inject.internal.InjectorImpl.getJustInTimeBinding(InjectorImpl.java:233)  
com.google.inject.internal.InjectorImpl.getBindingOrThrow(InjectorImpl.java:184)  
com.google.inject.internal.InjectorImpl.getInternalFactory(InjectorImpl.java:818)  
com.google.inject.internal.InjectorImpl.getProviderOrThrow(InjectorImpl.java:932)  
com.google.inject.internal.InjectorImpl.getProvider(InjectorImpl.java:965)  
com.google.inject.internal.InjectorImpl.getProvider(InjectorImpl.java:926)  
com.google.inject.internal.InjectorImpl.getInstance(InjectorImpl.java:978)  
org.sonatype.guice.bean.reflect.AbstractDeferredClass.get(AbstractDeferredClass.java:48)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.ProviderToInternalFactoryAdapter\$1.call(ProviderToInternalFactoryAdapter.java:46)  
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)  
com.google.inject.internal.ProviderToInternalFactoryAdapter.get(ProviderToInternalFactoryAdapter.java:40)  
com.google.inject.Scopes\$1\$1.get(Scopes.java:63)  
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)  
com.google.inject.internal.InjectorImpl\$4\$1.call(InjectorImpl.java:943)

```
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:939)
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)
org.sonatype.guice.plexus.binders.PlexusRequirements$RequirementProvider.get(PlexusRequirements.java:222)
org.sonatype.guice.plexus.binders.ProvidedPropertyBinding.injectProperty(ProvidedPropertyBinding.java:50)
org.sonatype.guice.bean.inject.BeanInjector.injectMembers(BeanInjector.java:53)
com.google.inject.internal.MembersInjectorImpl.injectMembers(MembersInjectorImpl.java:120)
com.google.inject.internal.ConstructorInjector.construct(ConstructorInjector.java:94)
com.google.inject.internal.ConstructorBindingImpl$Factory.get(ConstructorBindingImpl.java:253)
com.google.inject.internal.InjectorImpl$4$1.call(InjectorImpl.java:943)
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:939)
com.google.inject.internal.InjectorImpl.getInstance(InjectorImpl.java:978)
org.sonatype.guice.bean.reflect.AbstractDeferredClass.get(AbstractDeferredClass.java:48)
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)
com.google.inject.internal.ProviderToInternalFactoryAdapter$1.call(ProviderToInternalFactoryAdapter.java:46)
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:996)
com.google.inject.internal.ProviderToInternalFactoryAdapter.get(ProviderToInternalFactoryAdapter.java:40)
com.google.inject.Scopes$1$1.get(Scopes.java:63)
com.google.inject.internal.InternalFactoryToProviderAdapter.get(InternalFactoryToProviderAdapter.java:40)
com.google.inject.internal.InjectorImpl$4$1.call(InjectorImpl.java:943)
com.google.inject.internal.InjectorImpl.callInContext(InjectorImpl.java:989)
com.google.inject.internal.InjectorImpl$4.get(InjectorImpl.java:939)
org.sonatype.guice.bean.locators.QualifiedBean.getValue(QualifiedBean.java:85)
org.sonatype.guice.plexus.locators.LazyPlexusBean.getValue(LazyPlexusBean.java:55)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:234)
org.codehaus.plexus.DefaultPlexusContainer.lookup(DefaultPlexusContainer.java:214)
org.sonatype.plexus.rest.PlexusServerServlet.createApplication(PlexusServerServlet.java:49)
com.noelios.restlet.ext.servlet.ServerServlet.getApplication(ServerServlet.java:703)
com.noelios.restlet.ext.servlet.ServerServlet.init(ServerServlet.java:818)
javax.servlet.GenericServlet.init(GenericServlet.java:212)
org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:852)
org.apache.coyote.http11.Http11Protocol$Http11ConnectionHandler.process(Http11Protocol.java:588)
org.apache.tomcat.util.net.JIoEndpoint$Worker.run(JIoEndpoint.java:489)
java.lang.Thread.run(Thread.java:595)
```

弄了好长时间,网上也没有答案,希望您能帮助我一下...

27 楼 [hantsy](#) 2009-07-06

这个开放版本的试用过，相对其它repository mananger工具没有太大的优势。

pro我很喜欢的一个特性，staging，这个要收费才行。

nexus book 中后面几篇都是基于pro版本的。

希望 maven book 的翻译，nexus章节部分已经从 maven book 抽出成为 nexus book，能够保持在抽出前的版本。

26 楼 [juvenshun](#) 2009-06-10  
pinghe 写道

juvenshun 写道pinghe 写道还发现一个问题，使用pro版本，集成LDAP于windows 活动目录，用户的主要组在Selected Role中没有罗列出来pro可是要收费的，仅玩玩试用版，还是你想买阿？试用一下。前面帖子发现中文支持也有问题

Nexus暂时还没有考虑国际化哦  
25 楼 [pinghe](#) 2009-06-09  
juvenshun 写道

pinghe 写道还发现一个问题，使用pro版本，集成LDAP于windows 活动目录，用户的主要组在Selected Role中没有罗列出来pro可是要收费的，仅玩玩试用版，还是你想买阿？

试用一下。前面帖子发现中文支持也有问题  
24 楼 [juvenshun](#) 2009-06-09  
pinghe 写道

还发现一个问题，使用pro版本，集成LDAP于windows 活动目录，用户的主要组在Selected Role中没有罗列出来

pro可是要收费的，仅玩玩试用版，还是你想买阿？  
« 上一页 1 [2](#) [3](#) [下一页](#) »

发表评论



[您还没有登录,请您登录后再发表评论](#)