

打项目jar包到Nexus私库上

打项目jar包到Nexus私库上不是程序员经常性的操作，所以很多程序员对这里不是很清晰，今天刚才尝试了这里，把实战操作的过程和结果记录下来

下面我们使用这两个项目demo_parent、demo_springboot_rmq演示利用maven插件release:prepare,release:perform打项目jar包到Nexus私库上

前提条件：

1. Nexus

Nexus私库已经搭建好了

2. intelliJ

假设我们用的intelliJ操作这个过程

3. maven

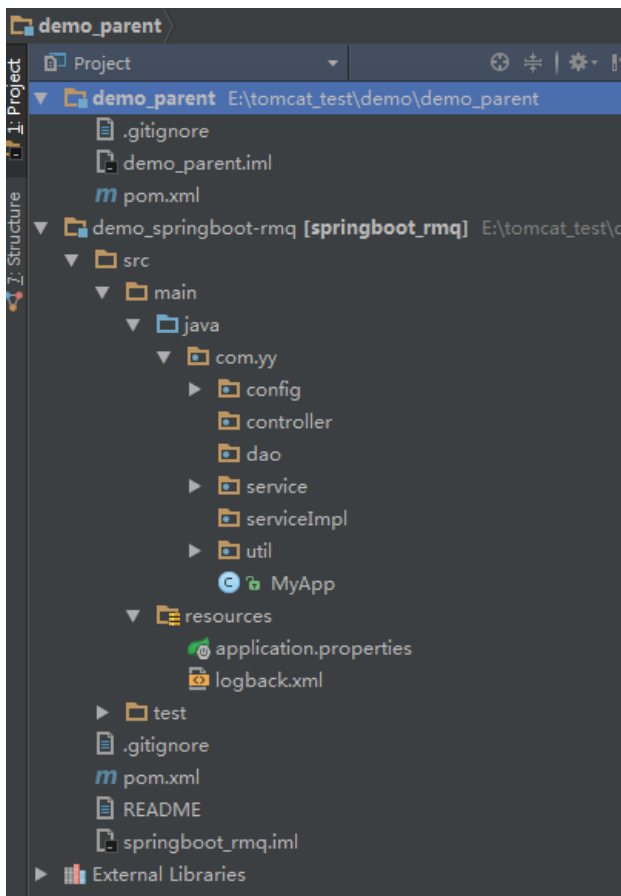
下载maven文件(我用的版本：apache-maven-3.3.9)，设置maven文件中的setting.xml文件

添加如下内容

```
<server>
  <id>yy-Releases</id>
  <username>Nexus用户名</username>
  <password>Nexus用户密码</password>
</server>
<server>
  <id>yy-Snapshots</id>
  <username>Nexus用户名</username>
  <password>Nexus用户密码</password>
</server>

<mirrors>
<mirror>
  <id>nexus</id>
  <mirrorOf>*</mirrorOf>
  <url>http://186.11.27.88:8081/nexus/content/groups/public/</url>
</mirror>
</mirrors>
```

项目结构：demo_springboot_rmq(子)依赖demo_parent(父)



>>> 打包并发布demo_parent项目

重点看下两个项目的pom文件

demo_parent的pom.xml

当前版本

```
<groupId>com. yy</groupId>  
💡 <artifactId>demo_parent</artifactId>  
<version>1. 0. 2-SNAPSHOT</version>  
<packaging>pom</packaging>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-release-plugin</artifactId>
      <version>2.0-beta-7</version>
      <configuration>
        <providerImplementations>
          <git>jgit</git>
        </providerImplementations>
        <username>git 远程库网站(如github)用户名</username>
        <password>git 远程库网站(如github)密码</password>
        <tagBase>${project.artifactId}-${project.version}</tagBase>
        <!--<goals>-f ${project.artifactId}/pom.xml deploy</goals>-->
        <goals>-f pom.xml deploy</goals>
      </configuration>
      <dependencies>
        <dependency>
          <groupId>org.apache.maven.scm</groupId>
          <artifactId>maven-scm-provider-jgit</artifactId>
          <version>1.9.5</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
</build>
```

```

<repositories>
  <repository>
    <id>nexus</id>
    <name>[redacted]</name>
    <url>http://私库的ip:端口/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
      <updatePolicy>always</updatePolicy>
    </snapshots>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>nexus</id>
    <name>[redacted]</name>
    <url>http://私库的ip:端口/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
      <updatePolicy>always</updatePolicy>
    </snapshots>
  </pluginRepository>
</pluginRepositories>

```

```

<distributionManagement>
  <repository>
    <id>对应maven的setting文件的server元素的id元素值</id>
    <name>对应maven的setting文件的server元素的名字元素值</name>
    <url>http://私库的ip:端口/nexus/content/repositories/releases</url>
  </repository>
  <snapshotRepository>
    <id>对应maven的setting文件的server元素的id元素值</id>
    <name>对应maven的setting文件的server元素的名字元素值</name>
    <url>http://私库的ip:端口/nexus/content/repositories/snapshots/</url>
  </snapshotRepository>
</distributionManagement>

<!--maven scm配置-->
<scm>
  <connection>scm:git:你项目所在的git路径</connection>
  <developerConnection>scm:git:你项目所在的git路径</developerConnection>
  <url>你项目所在的git路径/tree/master</url>
</scm>

```

下面是build.plugins和repositories、pluginRepositories、distributionManagement、scm这几个元素的实战例子

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>

    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-release-plugin</artifactId>
      <version>2.0-beta-7</version>
      <configuration>
        <providerImplementations>
          <git>jgit</git>
        </providerImplementations>
        <username>skyler</username>
        <password>123456</password>
        <tagBase>${project.artifactId}-${project.version}</tagBase>
        <!--<goals>-f ${project.artifactId}/pom.xml deploy</goals>-->
        <goals>-f pom.xml deploy</goals>
      </configuration>
      <dependencies>
        <dependency>
          <groupId>org.apache.maven.scm</groupId>
          <artifactId>maven-scm-provider-jgit</artifactId>
          <version>1.9.5</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
</build>
```

```

<repositories>
  <repository>
    <id>nexus</id>
    <name>local private nexus</name>
    <url>http://186.11.27.88:8081/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
      <updatePolicy>always</updatePolicy>
    </snapshots>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>nexus</id>
    <name>local private nexus</name>
    <url>http://186.11.27.88:8081/nexus/content/groups/public/</url>
    <releases>
      <enabled>true</enabled>
    </releases>
    <snapshots>
      <enabled>true</enabled>
      <updatePolicy>always</updatePolicy>
    </snapshots>
  </pluginRepository>
</pluginRepositories>

```

```

<distributionManagement>
  <repository>
    <id>yy-Releases</id>
    <name>yy-Releases</name>
    <url>http://186.11.27.88:8081/nexus/content/repositories/releases</url>
  </repository>
  <snapshotRepository>
    <id>yy-Snapshots</id>
    <name>yy-Snapshots</name>
    <url>http://186.11.27.88:8081/nexus/content/repositories/snapshots/</url>
  </snapshotRepository>
</distributionManagement>

<!--maven scm配置-->
<scm>
  <connection>scm:git:http://186.11.27.88:8888/skyler/my_test_parent.git</connection>
  <developerConnection>scm:git:http://186.11.27.88:8888/skyler/my_test_parent.git</developerConnection>
  <url>http://186.11.27.88:8888/skyler/my_test_parent/tree/master</url>
</scm>

```

待maven的setting.xml文件和demo_parent的pom.xml文件配置完了，我们即将运行maven的release:prepare命令。

注意，运行命令前，本地所有的修改都要push到远程库上，否则运行下面的命令会报错。

开始运行命令：

> release:prepare-->生成tag(远程和本地都生成，升级本地pom version版本号)

双击intellij maven插件的release:prepare命令

观察日志：

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building demo_parent 1.0.2-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-release-plugin:2.0-beta-7:prepare (default-cli) @ demo_parent ---
[INFO] Change the default 'git' provider implementation to 'jgit'.
[INFO] Verifying that there are no local modifications...
[INFO] Checking dependencies and plugins for snapshots ...
What is the release version for "demo_parent"? (com.yy:demo_parent) 1.0.2: :
What is SCM release tag or label for "demo_parent"? (com.yy:demo_parent) demo_parent-1.0.2: :
What is the new development version for "demo_parent"? (com.yy:demo_parent) 1.0.3-SNAPSHOT: : |
```

这三步可以让你手动输入发布的jar包名，一般默认就行，直接敲回车，继续

```
[INFO] Release preparation complete.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:58 min
[INFO] Finished at: 2017-03-02T18:36:58+08:00
[INFO] Final Memory: 19M/199M
[INFO] -----
```

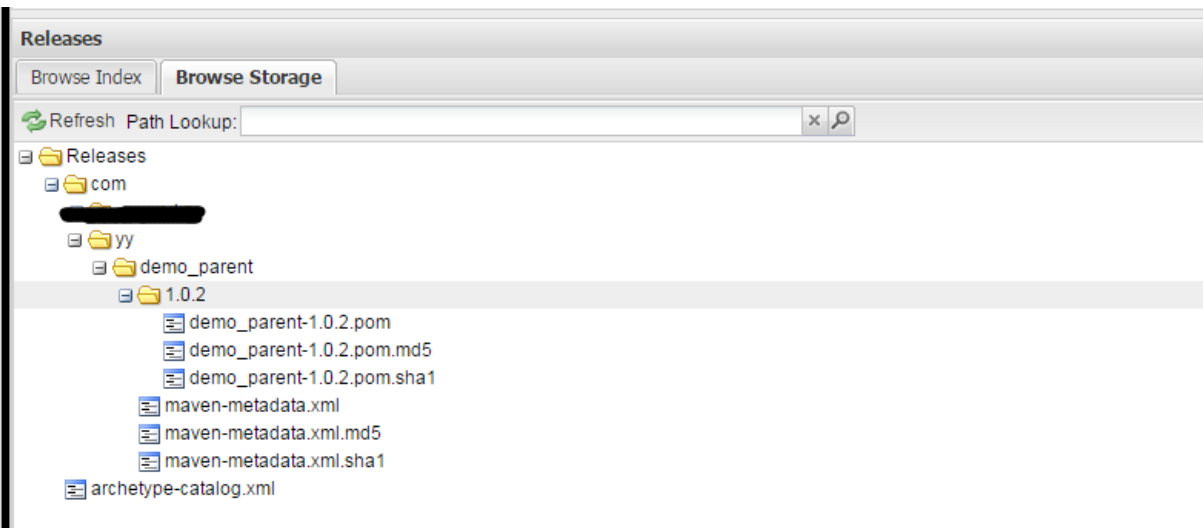
到这说明release:prepare运行成功了，这时你可以去远程库看看了，你会发现多了一个tag，而且tag项目的pom.xml文件的version元素由<version>1.0.2-SNAPSHOT</version>变成了<version>1.0.2</version>，同时，你本地项目的pom.xml中的由<version>1.0.2-SNAPSHOT</version>变成了<version>1.0.3-SNAPSHOT</version>

> release:perform-->把tag打成jar包push到Nexus私库中

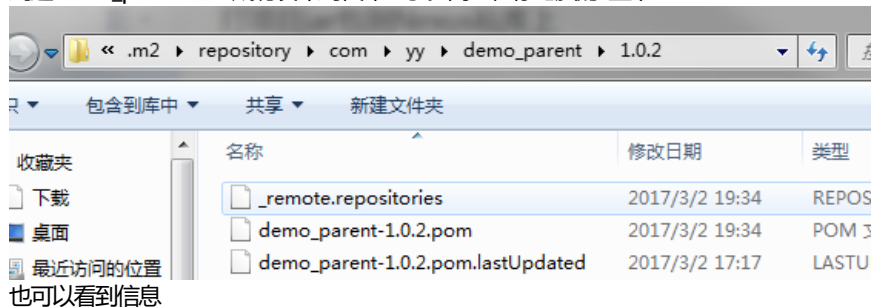
接着，运行release:perform，查看日志信息

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.425 s
[INFO] Finished at: 2017-03-02T19:34:34+08:00
[INFO] Final Memory: 27M/296M
[INFO] -----
[INFO] Cleaning up after release...
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 9.684 s
[INFO] Finished at: 2017-03-02T19:34:35+08:00
[INFO] Final Memory: 15M/220M
[INFO] -----
```

日志显示成功了，这时去Nexus私库看一下，效果如下图

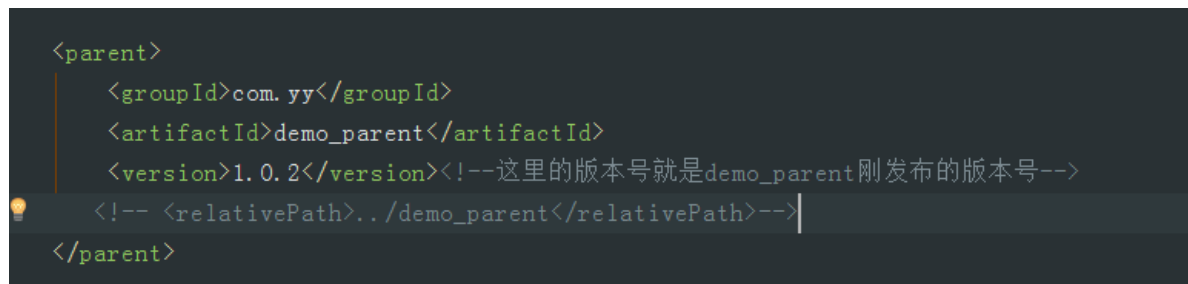


到这demo_parent已经成功发布到私库上了。而且在你电脑的c盘下



>>> 打包并发布demo_springboot_rmj项目

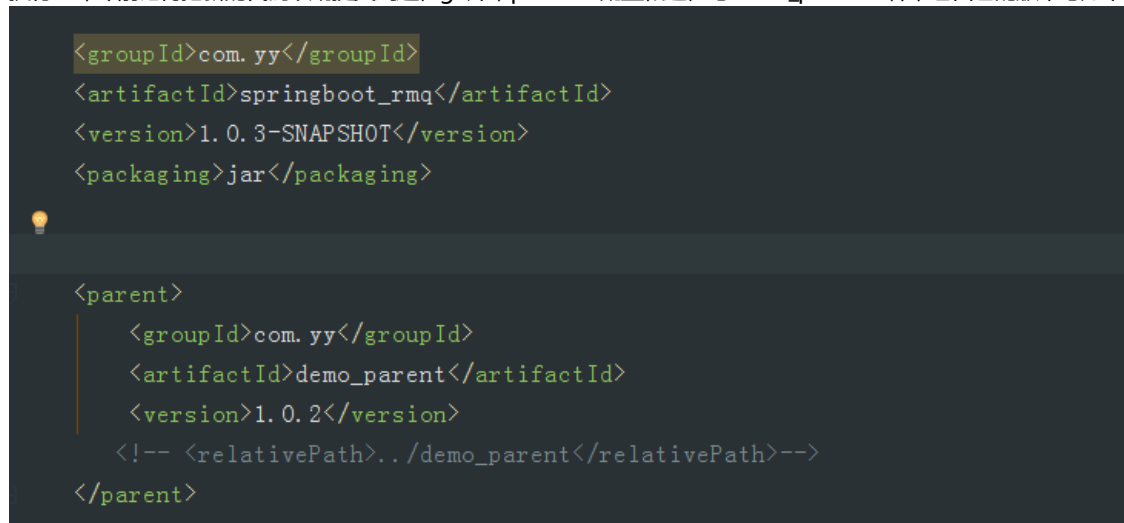
从demo_springboot_rmj的pom.xml可以看出与demo_parent的关系



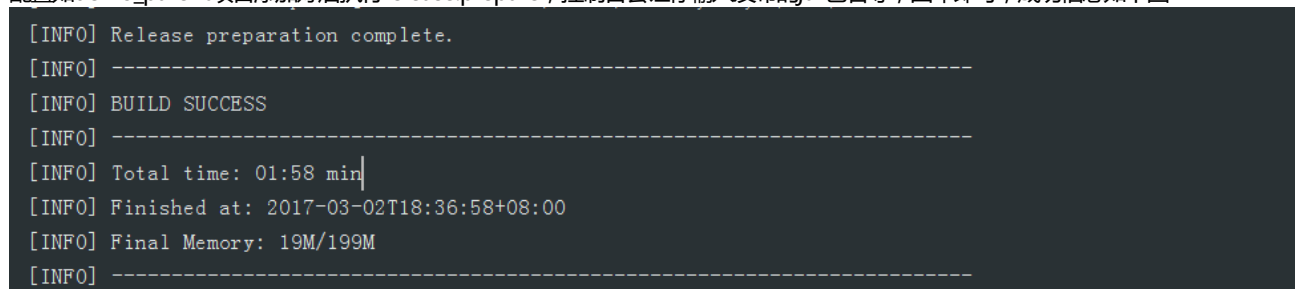
在打包发布demo_springboot_rmj前上图的版本号要指定明确为demo_parent发布成功的版本号中的一个，如上图

> release:prepare-->生成tag(远程和本地都生成，升级本地pom version版本号)

执行此命令前记得把新的代码改动提交到远程git库，pom.xml配置和过程与demo_parent一样，它自己的版本号如下图



配置如demo_parent项目添加好后执行release:prepare，控制台会让你输入发布的jar包名等，回车即可，成功信息如下图



接着执行下一个命令

> release:perform-->把tag打成jar包push到Nexus私库中

成功信息如下图


```
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 4.425 s  
[INFO] Finished at: 2017-03-02T19:34:34+08:00  
[INFO] Final Memory: 27M/296M  
[INFO] -----  
[INFO] Cleaning up after release...  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 9.684 s  
[INFO] Finished at: 2017-03-02T19:34:35+08:00  
[INFO] Final Memory: 15M/220M  
[INFO] -----
```

这时去Nexus私库看一下，你会发现刚打的jar出现在Nexus私库中