1、部署自己的项目853769620

我们已经可以让公司内所有项目强制全部走私服了，接下来，就是如果开发好某个模块之后，需要将其发布到私服上去，供其他的项目使用了。

（1）发布仓库配置

将项目发布包部署到哪个仓库中，是需要用下面的pom.xml中的配置来设置的：

<distributionManagement>

<repository>

<id> nexus-releases</id>

<name> Nexus Release Repository</name>

<url>http://localhost:8081/nexus/content/repositories/releases/</url>

</repository>

<snapshotRepository>

<id> nexus-snapshots</id>

<name> Nexus Snapshot Repository</name>

<url>http://localhost:8081/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

（2）部署专用账号的配置

但是nexus仓库对于普通的匿名用户是只读的，也就是说，只能下载依赖，不能部署发布包，因此如果要能够部署发布包，还需要在settings.xml文件里通过<server>元素配置使用专用的部署用户，来通过认证，进行发布包的部署。

需要在settings中配置：

<servers>

<server>

<id>nexus-releases</id>

<username>deployment</username>

<password>deployment123</password>

</server>

<server>

<id>nexus-snapshots</id>

<username>deployment</username>

<password>deployment123</password>

</server>

</servers>

（3）执行Maven deploy命令部署到私服

clean deploy这个命令，执行maven命令以后会发生什么事情，我们目前先不用管他，因为依赖管理+仓库讲解完之后，我们马上就要讲解maven的命令背后的深入的原理，在干嘛。。。

你只要知道，我们接下来去执行mvn clean deploy命令，就可以让maven自动给我们编译源代码、运行单元测试、打成jar包、将jar包安装到本地仓库、将Jar包部署到配置的远程私服仓库里面去。

是这样的，在私服上面，如果上传snapshot版本的jar包呢，在私服上存储的时候，会自动带上一个时间戳。这个主要是因为考虑到在开发过程中会频繁的部署snapshot包，所以会用时间戳来区分一下。但是这个对我们是透明的，如果我们团队里其他哥儿们要用你的这个snapshot jar包，直接还是用你的坐标来声明依赖就可以了，他是不用去考虑这个时间戳的问题的，私服会替你管理好。

（4）mvn命令的说明

mvn clean package：清理、编译、测试、打包

mvn clean install：清理、编译、测试、打包、安装到本地仓库，比如你自己负责了3个工程的开发，互相之间有依赖，那么如果你开发好其中一个工程，需要在另外一个工程中引用它，此时就需要将开发好的工程jar包安装到本地仓库，然后才可以在另外一个工程声明对它的依赖，此时会直接取用本地仓库中的jar包

mvn clean deploy：清理、编译、测试、打包、安装到本地仓库、部署到远程私服仓库，这个其实就是你负责的工程写好了部分代码，别人需要依赖你的jar包中提供的接口来写代码和测试。此时你需要将snapshot jar包发布到私服的maven-snapshots仓库中。供别人在本地声明对你的依赖和使用。

2、对于极其少数的jar包手动上传

有些第三方厂商的jar包，比如特殊数据库厂商的jdbc驱动，或者某个厂商的支付相关的api jar包，是不提供公网下载的，只能从他们那里拿到一个jar包，此时就需要手动将其上传到3rd party仓库里面去。

新版本里面，其实主要是建议用命令行手动上传的方式，就不用界面上上传的方式了：

选择3rd party，选择Artifact Upload，上传即可

<server>

<id>nexus-3rd-party</id>

<username>deployment</username>

<password>deployment123</password>

</server>

mvn deploy:deploy-file -DgroupId=com.csource -DartifactId=fastdfs-client-java -Dversion=1.24 -Dpackaging=jar -Dfile=F:\DevelopmentKit\fastdfs\_client\_v1.24.jar -Durl=http://localhost:8081/repository/3rd-party/ -DrepositoryId=nexus-3rd-party