**每日作业卷答案**

**maven第2天**

极客营

# 关卡1

## 训练案例1

### 训练描述

项目发布到nexus私服

### 操作步骤答案

1. 配置maven的settings.xml文件用于发布的用户名和密码；

在maven的conf目录下的settings.xml文件中的<servers>标签里面添加如下内容：

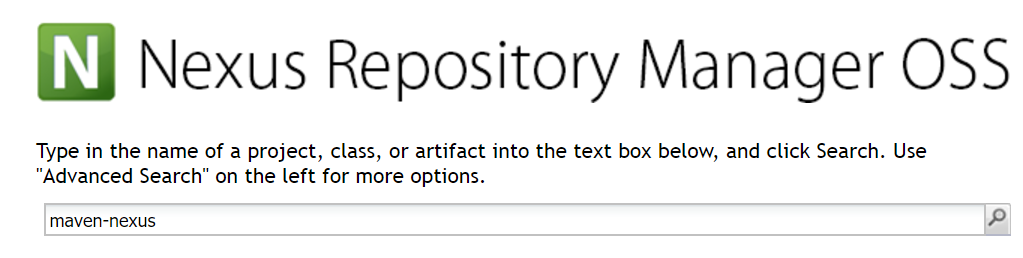
|  |
| --- |
| <server>  <id>releases</id>  <username>admin</username>  <password>admin123</password>  </server>  <server>  <id>snapshots</id>  <username>admin</username>  <password>admin123</password>  </server> |

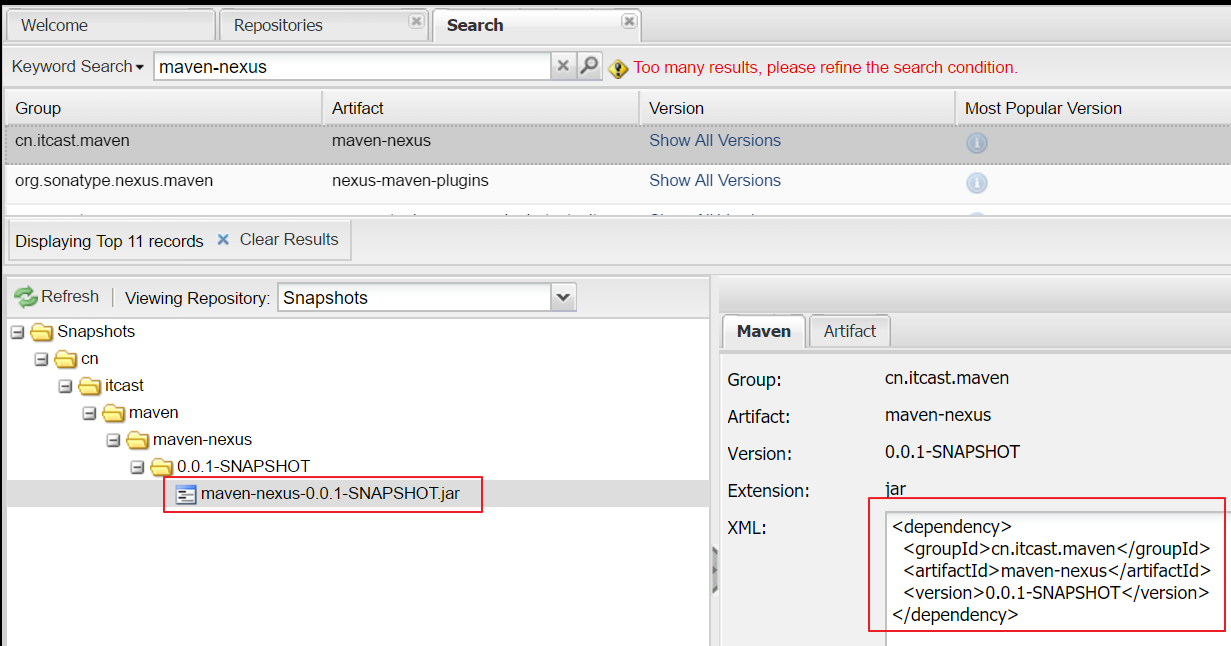
1. 创建maven-nexus的java工程；
2. 配置maven-nexus工程的pom.xml文件，添加发布的路径；

Pom.xml文件中添加如下配置：

|  |
| --- |
| <!-- 构建部署项目 -->  <distributionManagement>  <repository>  <id>releases</id>  <name>Internal Releases</name>  <url>http://localhost:8081/nexus/content/repositories/releases/</url>  </repository>  <snapshotRepository>  <id>snapshots</id>  <name>Internal Snapshots</name>  <url>http://localhost:8081/nexus/content/repositories/snapshots/</url>  </snapshotRepository>  </distributionManagement> |

1. 在项目中右击选择run as 🡪 maven build… 然后在Goals中输入 deploy 将项目的包部署上传到nexus私服；
2. 访问nexus在主页面中搜索maven-nexus，如果发布成功则能找到具体的坐标（注意：如果需要将多次部署，需要在nexus私服中对对应的仓库的Configuration——》Development Policy设置“Allow Redeploy”）





# 关卡2

## 训练案例1

### 训练描述

nexus私服中上传jar包，项目中引用坐标下载jar

### 操作步骤答案

1. 在maven的settings.xml配置文件中配置私服仓库地址；

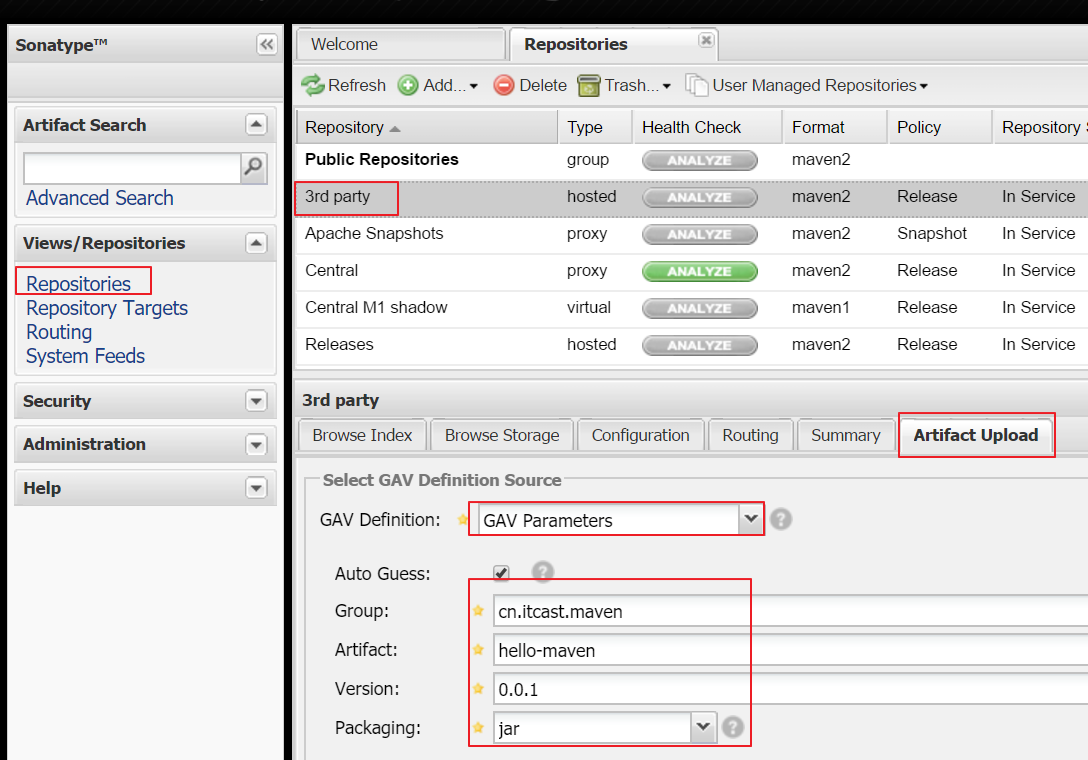
在settings.xml文件的<profiles>标签里面添加如下内容：

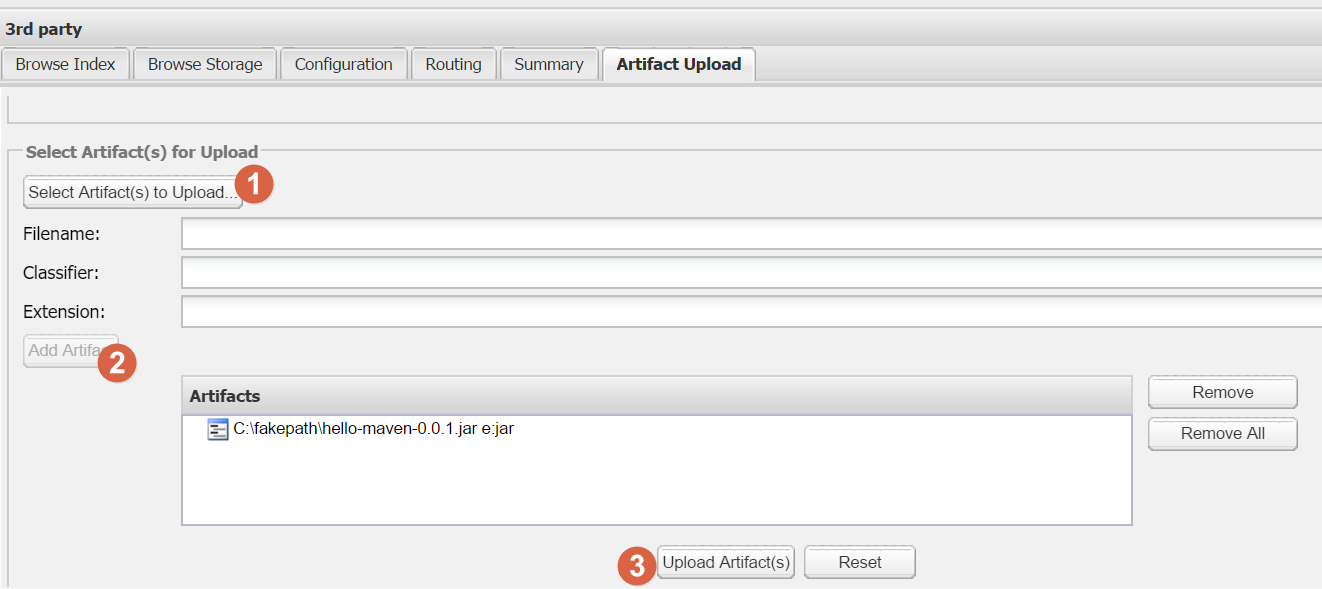
|  |
| --- |
| <profile>  <id>dev</id>  <repositories>  <repository>  <id>nexus</id>  <url>http://localhost:8081/nexus/content/groups/public/</url>  <releases><enabled>true</enabled></releases>  <snapshots><enabled>true</enabled></snapshots>  </repository>  </repositories>  <pluginRepositories>  <pluginRepository>  <id>nexus</id>  <url>http://localhost:8081/nexus/content/groups/public/</url>  <releases><enabled>true</enabled></releases>  <snapshots><enabled>true</enabled></snapshots>  </pluginRepository>  </pluginRepositories>  </profile> |

在settings.xml文件的</settings>标签之上添加如下内容：

|  |
| --- |
| <activeProfiles>  <!--make the profile active all the time -->  <activeProfile>dev</activeProfile>  </activeProfiles> |

1. 访问并登录nexus私服（帐号：admin，密码：admin123）；
2. 找到Repositories，然后选择“3rd party”仓库，点击“Attifact Upload”选项卡将第一天作业卷中的hello-maven/target/hello-maven-0.0.1.jar这个包上传到私服；





1. 打开maven-nexus工程的pom.xml文件，添加具体的hello-maven的依赖坐标（可以到本地仓库中将hello-maven的安装删除，让其在本地仓库中没有的情况下到私服去下载）

在maven-nexus的pom.xml文件中添加如下依赖信息：

|  |
| --- |
| <dependencies>  <dependency>  <groupId>com.igeek.maven</groupId>  <artifactId>hello-maven</artifactId>  <version>0.0.1</version>  </dependency>  </dependencies> |

# 关卡3

## 训练案例1

### 训练描述

使用命令打war包并将其部署到独立的tomcat下。

### 操作步骤答案

1. 使用简单方式创建maven web项目maven-web-2；（注意创建项目过程中跳过原型选择、打包方式选择war）
2. 在maven-web-2项目下添加/maven-web-2/src/main/webapp/WEB-INF/web.xml文件和添加/maven-web-2/src/main/webapp/index.html文件；

web.xml内如如下：

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* version=*"2.5"*>  <display-name>maven-web-2</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  </web-app> |

index.html内容如下：

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=*"UTF-8"*>  <title>Insert title here</title>  </head>  <body>  Hello Maven.  </body>  </html> |

1. 在eclipse中对maven-web-2项目右击并选择Rus as🡪Maven build…🡪在Goals里面输入package打包生成maven-web-2\target\maven-web-2-0.0.1-SNAPSHOT.war；
2. 将maven-web-2\target\maven-web-2-0.0.1-SNAPSHOT.war复制到任意一台tomcat的webapps目录下并启动tomcat；

1. 访问http://localhost:8080/maven-web-2-0.0.1-SNAPSHOT 查看是否能正确访问到index.html的内容。