

Maven02

笔记本: Maven
创建时间: 2018/10/15 22:29 更新时间: 2018/10/17 13:20
作者: 155307642@qq.com
URL: https://blog.csdn.net/hjh156/article/details/59635587?utm_source=blogxgwz1

一、Maven插件

Maven的核心仅仅定义了抽象的生命周期，具体的任务都是交由插件完成的，每个插件都能实现多个功能，每个功能就是一个插件目标。Maven的生命周期与插件目标相互绑定，以完成某个具体的构建任务。例如：compile就是插件maven-compiler-plugin的一个插件目标。

1、jdk编译插件

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.luban.maven</groupId>
  <artifactId>Hello</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.0</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
          <encoding>UTF-8</encoding>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

2、源码打包插件

1.解决idea中，创建web项目慢的方式：

在Properties中添加Name:archetypeCatalog和Value:internal,如下图那样

也可以在设置中指定Maven-Runne的VM: -DarchetypeCatalog为internal

2、打包插件pom.xml

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-source-plugin</artifactId>
  <version>2.2.1</version>

  <executions>
    <execution>
```

```

        <goals>
            <goal>jar-no-fork</goal>
        </goals>

        <phase>verify</phase>

    </execution>
</executions>
</plugin>

```

3、Tomcat插件

```

<!--Tomcat插件 -->
<plugin>

    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.1</version>

    <configuration>
        <port>8081</port>
        <server>tomcat7</server>
    </configuration>
</plugin>

```

二、继承与聚合

继承是为了消除重复，可以把pom中很多相同的配置提取出来，如：groupId，version等。在使用的时候，子工程直接继承父工程的依赖版本号，子工程中不再需要指定具体版本号，方便统一管控项目的依赖版本问题。

- 父工程的设置
 - 父工程的pom.xml中打包的方式必须设置为pom方式。
 -

1、父工厂的作用

2、聚合工程的作用

如果想一次构建多个项目模块，那则需要对多个项目模块进行聚合

```

<modules>
    <module>子项目名称1</module>
    <module>子项目名称2</module>
    <module>子项目名称3</module>
</modules>

```

3、继承与聚合

聚合主要是为了快速构建项目；继承主要是为了消除重复。

三、Maven私服

1、私服的好处

- 缓存Maven中央仓库的jar包，这样不需要每次本地仓库没有jar包而不必到中央仓库下载，而是到私服下载。
- 解决公司无法上网，而又要连接中央仓库的问题。只要连接私服，确保私服可以连接到中央仓库。
- 方便公司内部不同团队或者项目共享jar包，需要共享jar包，可以上传到私服，通过私服共享。

2、Nexus安装与仓库概念

- Nexus下载地址：<https://www.sonatype.com/nexus-repository-oss>
- 安装
 - 解压文件
 - 启动服务 nexus.exe /run （启动成功后不要关闭命令行窗口）

- 安装服务 `nexus.exe /install` (重新使用新的命令行安装, 可能出现权限不足, 请以管理员身份运行命令行)
- 启动服务 `net start nexus`
- 停止服务 `net stop nexus`
- 访问页面 <http://localhost:8081/>
- 初始登录用户名和密码: `admin/admin123`

3、从Nexus私服下载资源

找到maven配置文件`settings.xml`, 添加如下配置

1) 添加镜像配置, 将所有访问外网仓库的请求指向私服

```
<mirror>
  <id>nexus</id>
  <mirrorOf>*</mirrorOf>
  <url>http://localhost:8081/repository/maven-public/</url>
</mirror>
```

2) 添加profiles, 所有请求均通过镜像

```
<profile>
  <id>nexus</id>
  <repositories>
    <repository>
      <id>central</id>
      <url>https://repo1.maven.org/maven2</url>
      <releases><enabled>true</enabled></releases>
      <snapshots><enabled>true</enabled></snapshots>
    </repository>
  </repositories>
  <pluginRepositories>
    <pluginRepository>
      <id>central</id>
      <url>https://repo1.maven.org/maven2</url>
      <releases><enabled>true</enabled></releases>
      <snapshots><enabled>true</enabled></snapshots>
    </pluginRepository>
  </pluginRepositories>
</profile>
```

3) 生效配置

```
<activeProfiles>
  <activeProfile>nexus</activeProfile>
</activeProfiles>
```

4、上传资源到Nexus私服

1、在`pom.xml`中配置, 上传资源到nexus

```
<distributionManagement>
  <repository>
    <id>release</id>
    <name>Internal Release</name>
    <url>http://localhost:8081/repository/maven-releases/</url>
  </repository>
  <snapshotRepository>
    <id>snapshots</id>
    <name>Internal Snapshots</name>
    <url>http://localhost:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

2、在maven的配置文件中配置上传权限

```
<server>
  <id>release</id>
  <username>admin</username>
  <password>admin123</password>
</server>
<server>
  <id>snapshots</id>
  <username>deployment</username>
  <password>deployment123</password>
</server>
```

作业：

- 1、使用Tomcat插件完成web的发布。
- 2、创建2个项目，公共部分放到父工程中。（继承的使用）