



http://mushiqianmeng.blog.5lcto.com 【复制】 【订阅】

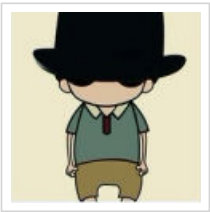
博 客 | 写博文 | 帮 助

首页 | nocloud | maven | log | bugtracker | 虚拟化 | linux | JAVA基础入门 | Java Web | spring | algorithm | 微新闻 | 工作生活 | hibernate | 自娱自乐 | asm | libvirt | eclipse | junit | Netty | 翻译

mushiqianmeng 的BLOG

相关视频课程

更多



写留言

去学院学习

发消息

加好友链接

进家园 加好友

博客统计信息

51CTO推荐博客

用户名: mushiqianmeng

文章数: 63

评论数: 352

访问量: 1000598

无忧币: 6725

博客积分: 3042

博客等级: 7

注册日期: 2011-11-21

热门专题

更多>>



每天5分钟玩转  
OpenStack  
阅读量: 5863



【51CTO三周年】我在  
学院不得不说的收获  
阅读量: 12276



从菜鸟到老鸟-教你玩  
转Mac操作系统  
阅读量: 416873



QT学习之路: 从入门到  
精通  
阅读量: 1104361

热门文章

Win7 64位环境下JDK和Ecl..

教你打包Java程序, jar..

Spring配置项<context..



Maven 管理主流框架依  
赖视频课程第4季(共15  
500人学习



Maven 管理主流框架依  
赖视频课程第3季(共10  
540人学习



Maven 管理主流框架依  
赖视频课程第1季(共19  
1191人学习

博主的更多文章>>

转载

Maven中<dependencies>节点和<dependencyManagement>节点的区别

2011-11-21 13:09:47

标签: maven dependencies dependencyManagement 休闲 职场

1 . 使用项目继承

利用项目继承可以将结构信息 , 部署信息 , 共同的依赖信息放置在单一的位置。在每个工程的 pom 中 :

```
01. <parent>
02.
03. <groupId>org.apache.maven.proficio</groupId>
04.
05. <artifactId>proficio</artifactId>
06.
07. <version>1.0-SNAPSHOT</version>
08.
09. </parent>
```

这使得项目的 pom 可以继承顶层 pom 中的定义 , 检查顶层 pom 的 dependencies 部分 :

```
01. <project>
02.
03. <dependencies >
04.
05. <dependency >
06.
07. <groupId>junit</groupId>
08.
09. <artifactId>junit</artifactId>
10.
11. <version>4.7</version>
12.
13. <scope>test</scope>
14.
15. </dependency >
16.
17. </dependencies >
18.
19. </project>
```

在各个子模块的 pom 中没有对 Junit 依赖的定义 , 但是从顶层 pom 中继承了依赖的定义。

为了看清楚可以在一个子模块 pom 所在目录下 , 执行命令

```
01. #mvn help:effective-pom
```

可以看到最终起效果的 pom , 这在找错时很有效。

2 . 管理依赖

在 pom 中指明 dependency management 元素的方式 maven 结合项目继承来管理依赖。在多模块应用中 , 可能多个子项目会有共同的依赖。此时为了正确运行 , 必须让所有的子项目使用依赖项的同一版本。必须确保应用的各个项目

slf4j+logback配置方式, ..

原创Netty学习系列导航

Maven中<dependencies>..

Eclipse使用技巧 - 自定..

教你快速写出多线程JUnit..

搜索BLOG文章

搜索

最近访客

hello..

lxr215

68444355

羽光..

wx591..

qq591..

zzqvxzzq

NFYZ

懒阳..

htliuiv

gener..

王刚001

最新评论

JingXianSen: 楼主, 有好多问题想请教你。我没有..

horusa: 遇到同样的问题, 求解

wk117: 回复 sehui3: 同问啊..

阿莱士多拉: 不是四则运算不带空格, 请仔细阅读..

sehui3: 你好 我想问下 引用了第三方jar包 ..

51CTO推荐博文

更多&gt;&gt;

Mysql数据库连接查询

MySQL导入导出方法总结

mysql表碎片的查询自己回收

在CentOS 7上使用RPM包安装MySQL 5.7

Docker编排工具之Rancher-Server..

oracle如何修改单个用户密码永不过期

MariaDB10.1.22 Spider3.3腾讯补..

Zabbix监控MySQL小记

解决Linux下MySQL启动错误Startin..

sql优化-检索及定位

mysqldump备份与恢复 之 黑锅侠

友情链接

的依赖项和版本一致, 才能保证测试的和发布的是相同的成果。因此, 应在顶层的 pom 中定义共同的依赖关系。

在 Proficio 应用的顶层 pom 中的 dependency management 段如下：

```
01. <dependencyManagement>
02.
03. <dependencies>
04.
05. <dependency>
06.
07. <groupId>com.devzuz.mvnbook.proficio</groupId>
08.
09. <artifactId>proficio-model</artifactId>
10.
11. <version>${project.version}</version>
12.
13. </dependency>
14.
15. <dependency>
16.
17. <groupId>com.devzuz.mvnbook.proficio</groupId>
18.
19. <artifactId>proficio-api</artifactId>
20.
21. <version>${project.version}</version>
22.
23. </dependency>
24.
25. <dependency>
26.
27. <groupId>com.devzuz.mvnbook.proficio</groupId>
28.
29. <artifactId>proficio-core</artifactId>
30.
31. <version>${project.version}</version>
32.
33. </dependency>
34.
35. <dependency>
36.
37. <groupId>com.devzuz.mvnbook.proficio</groupId>
38.
39. <artifactId>proficio-store-memory</artifactId>
40.
41. <version>${project.version}</version>
42.
43. </dependency>
44.
45. <dependency>
46.
47. <groupId>com.devzuz.mvnbook.proficio</groupId>
48.
49. <artifactId>proficio-store-xstream</artifactId>
50.
51. <version>${project.version}</version>
52.
53. </dependency>
54.
55. <dependency>
56.
57. <groupId>org.codehaus.plexus</groupId>
58.
59. <artifactId>plexus-container-default</artifactId>
60.
61. <version>1.0-alpha-9</version>
62.
63. </dependency>
64.
65. </dependencies>
66.
67. </dependencyManagement>
```

注意 \${project.version} 变量指的是应用的 version 。

顶层 pom 中的 dependencies 与 dependencyManagement 中的 dependencies 元素有一个重要的区别：

dependencyManagement 中的 dependencies 元素只表明依赖项版本的优先选择，并不影响项目的依赖项；而 dependencies 元素则影响项目的依赖项。

检查 Proficio api 模块的 pom:

```
01. <project>
02.
03. <parent>
04.
05. <groupId>com.devzuz.mvnbook.proficio</groupId>
06.
07. <artifactId>proficio</artifactId>
```

IT精品课程
ERDP技术架构
雨中的海
8无聊
Coder论坛，爱好者..
被测试的小强 
运维笔记
我的One Coder博客
小任的学习笔记
煮酒品茶
网络之路
Seven的技术日志
桌面虚拟化杂谈
滴水穿石 
技术人才招聘

```
08.         <version>1.0-SNAPSHOT</version>
09.
10.
11.     </parent>
12.
13.     <modelVersion>4.0.0</modelVersion>
14.
15.     <artifactId>proficio-api</artifactId>
16.
17.     <packaging>jar</packaging>
18.
19.     <name>Proficio API</name>
20.
21.     <dependencies >
22.
23.         <dependency >
24.
25.             <groupId>com.devzuz.mvnbook.proficio</groupId>
26.
27.             <artifactId>proficio-model</artifactId>
28.
29.         </dependency >
30.
31.     </dependencies >
32.
33. </project>
```

其中没有指明依赖项的版本信息，在顶层 pom 中的 dependencyManagement 中表明其优选的版本是 \${project.version} 即 1.0-SNAPSHOT 。为使其模块 pom 完整，其版本信息会注入其中。

只有当 外层的dependencies 元素中没有指明版本信息时， dependencyManagement 中的 dependencies 元素才起作用。

注意：一个是项目依赖，一个是多模块maven项目时候的依赖管理控制的。

分享至:

收藏 

 0人 了这篇文章

类别: [maven](#) | 阅读 (29332) | 评论 (0) | [返回博主首页](#) | [返回博客首页](#)

[上一篇 nocloud开发随记-2011-11-20](#) [下一篇 maven依赖关系中 scope属性含义](#)

相关文章

- 掌握Maven
- maven打可执行的jar包以及classpath设置
- 解决一个Maven无法正常加载类库的问题
- Maven学习总结

职位推荐

- 游戏客户端开发工程师（cocos2dx）
- 深度学习算法工程师
- PHP研发工程师
- PowerBI/.NET 开发工程师
- Linux内核开发

文章评论

发表评论

昵 称

验证码:  [请点击后输入验证码](#) [博客过2级，无需填写验证码](#)

内 容

