Maven 环境配置 ➔

Maven 教程

Maven 翻译为"专家"、"内行"，是 Apache 下的一个纯 Java 开发的开源项目。基于项目对象模型（缩写：POM）概念，Maven利用一个中央信息片断能管理一个项目的构建、报告和文档等步骤。

Maven 是一个项目管理工具，可以对 Java 项目进行构建、依赖管理。

Maven 也可被用于构建和管理各种项目，例如 C#，Ruby，Scala 和其他语言编写的项目。Maven 曾是 Jakarta 项目的子项目，现为由 Apache 软件基金会主持的独立 Apache 项目。

阅读本教程前，您需要了解的知识

本教程主要针对初学者，帮助他们学习 Maven 工具的基本功能。完成本教程的学习后你的 Apache Maven 的专业知识将达到中等水平，随后你可以学习更高级的知识了。

阅读本教程，您需要有以下基础：Java 基础。

Maven 功能

Maven 能够帮助开发者完成以下工作：

- 构建
- 文档生成
- 报告
- 依赖
- SCMs
- 发布
- 分发
- 邮件列表

约定配置

Maven 提倡使用一个共同的标准目录结构，Maven 使用约定优于配置的原则，大家尽可能的遵守这样的目录结构。如下所示：

目录	目的
${basedir}	存放pom.xml和所有的子目录
${basedir}/src/main/java	项目的java源代码

目录	目的
${basedir}/src/main/resources	项目的资源，比如说property文件，springmvc.xml
${basedir}/src/test/java	项目的测试类，比如说Junit代码
${basedir}/src/test/resources	测试用的资源
${basedir}/src/main/webapp/WEB-INF	web应用文件目录，web项目的信息，比如存放web.xml、本地图片、jsp视图页面
${basedir}/target	打包输出目录
${basedir}/target/classes	编译输出目录
${basedir}/target/test-classes	测试编译输出目录
Test.java	Maven只会自动运行符合该命名规则的测试类
~/.m2/repository	Maven默认的本地仓库目录位置

Maven 特点

- 项目设置遵循统一的规则。
- 任意工程中共享。
- 依赖管理包括自动更新。
- 一个庞大且不断增长的库。
- 可扩展，能够轻松编写 Java 或脚本语言的插件。
- 只需很少或不需要额外配置即可即时访问新功能。
- **基于模型的构建** – Maven能够将任意数量的项目构建到预定义的输出类型中，如 JAR，WAR 或基于项目元数据的分发，而不需要在大多数情况下执行任何脚本。
- **项目信息的一致性站点** – 使用与构建过程相同的元数据，Maven 能够生成一个网站或PDF，包括您要添加的任何文档，并添加到关于项目开发状态的标准报告中。
- **发布管理和发布单独的输出** – Maven 将不需要额外的配置，就可以与源代码管理系统（如 Subversion 或 Git）集成，并可以基于某个标签管理项目的发布。它也可以将其发布到分发位置供其他项目使用。Maven 能够发布单独的输出，如 JAR，包含其他依赖和文档的归档，或者作为源代码发布。
- **向后兼容性** – 您可以很轻松的从旧版本 Maven 的多个模块移植到 Maven 3 中。
- 子项目使用父项目依赖时，正常情况子项目应该继承父项目依赖，无需使用版本号，

- **并行构建** – 编译的速度能普遍提高20 - 50 %。
- **更好的错误报告** – Maven 改进了错误报告，它为您提供了 Maven wiki 页面的链接，您可以点击链接查看错误的完整描述。

Maven 环境配置 →

1 篇笔记

写笔记

Maven 的 Snapshot 版本与 Release 版本

1、Snapshot 版本代表不稳定、尚处于开发中的版本。

2、Release 版本则代表稳定的版本。

3、什么情况下该用 SNAPSHOT?

协同开发时，如果 A 依赖构件 B，由于 B 会更新，B 应该使用 SNAPSHOT 来标识自己。这种做法的必要性可以反证如下：

- a. 如果 B 不用 SNAPSHOT，而是每次更新后都使用一个稳定的版本，那版本号就会升得太快，每天一升甚至每个小时一升，这就是对版本号的滥用。
- b.如果 B 不用 SNAPSHOT, 但一直使用一个单一的 Release 版本号，那当 B 更新后，A 可能并不会接受到更新。因为 A 所使用的 repository 一般不会频繁更新 release 版本的缓存（即本地 repository)，所以B以不换版本号的方式更新后，A在拿B时发现本地已有这个版本，就不会去远程 Repository下载最新的 B

4、 不用 Release 版本，在所有地方都用 SNAPSHOT 版本行不行？

不行。正式环境中不得使用 snapshot 版本的库。 比如说，今天你依赖某个 snapshot 版本的第三方库成功构建了自己的应用，明天再构建时可能就会失败，因为今晚第三方可能已经更新了它的 snapshot 库。你再次构建时，Maven 会去远程 repository 下载 snapshot 的最新版本，你构建时用的库就是新的 jar 文件了，这时正确性就很难保证了。

任人欺凌小师妹 6个月前 (09-30)