依赖这块已经彻底搞定了，然后就来看看maven的最重要的构建这块853769620

自动化的依赖管理

自动化的构建管理

mvn命令

mvn clean package：打包

mvn clean install：安装到本地

mvn clean deploy：部署到远程私服

1、一张图带你看懂maven生命周期以及执行原理

我之前见过很多人，讲解maven这块，一般都是参照那个maven实战那本书的组织结构去讲解的。那本书写的相当不错，但是并不能说就是学习maven的人最好的一个资料。讲解的顺序和组织，语言的表述，有的时候会让人觉得学习的也比较吃力。

上来就是maven生命周期，我就特别不喜欢那种讲解，我喜欢按照我自己的理解，站在学习人的角度，去考虑大家当前对maven是一种什么样的印象，理解到了什么程度，然后顺着你们当前的思路去讲解

先讲依赖，按照的工程开发的过程去讲解的，你先用maven搭建一个工程，依赖，开发代码，依赖管理机制，私服，仓库架构，部署到私服

maven生命周期，就是去解释mvn各种命令背后的原理

maven的生命周期，就是对传统软件项目构建工作的抽象

清理、初始化、编译、测试、打包、集成测试、验证、部署、站点生成

maven有三套完全独立的生命周期，clean，default和site。每套生命周期都可以独立运行，每个生命周期的运行都会包含多个phase，每个phase又是由各种插件的goal来完成的，一个插件的goal可以认为是一个功能。

这就是maven的生命周期 -> phase（可以理解为阶段） -> 插件的关系，也是maven构建执行的核心原理

你每次执行一个生命周期，都会依次执行这个生命周期内部的多个phase，每个phase执行时都会执行某个插件的goal完成具体的功能

2、maven的生命周期以及phase

clean生命周期包含的phase如下：

pre-clean

clean

post-clean

default生命周期包含的phase如下：

validate：校验这个项目的一些配置信息是否正确

initialize：初始化构建状态，比如设置一些属性，或者创建一些目录

generate-sources：自动生成一些源代码，然后包含在项目代码中一起编译

process-sources：处理源代码，比如做一些占位符的替换

generate-resources：生成资源文件，才是干的时我说的那些事情，主要是去处理各种xml、properties那种配置文件，去做一些配置文件里面占位符的替换

process-resources：将资源文件拷贝到目标目录中，方便后面打包

compile：编译项目的源代码

process-classes：处理编译后的代码文件，比如对java class进行字节码增强

generate-test-sources：自动化生成测试代码

process-test-sources：处理测试代码，比如过滤一些占位符

generate-test-resources：生成测试用的资源文件

process-test-resources：拷贝测试用的资源文件到目标目录中

test-compile：编译测试代码

process-test-classes：对编译后的测试代码进行处理，比如进行字节码增强

test：使用单元测试框架运行测试

prepare-package：在打包之前进行准备工作，比如处理package的版本号

package：将代码进行打包，比如jar包

pre-integration-test：在集成测试之前进行准备工作，比如建立好需要的环境

integration-test：将package部署到一个环境中以运行集成测试

post-integration-test：在集成测试之后执行一些操作，比如清理测试环境

verify：对package进行一些检查来确保质量过关

install：将package安装到本地仓库中，这样开发人员自己在本地就可以使用了

deploy：将package上传到远程仓库中，这样公司内其他开发人员也可以使用了

site生命周期的phase：

pre-site

site

post-site

site-deploy

3、默认的phase和plugin绑定

但是问题来了，那么我们直接运行mvn clean package的时候，每个phase都是由插件的goal来完成的，phase和plugin绑定关系是？

实际上，默认maven就绑定了一些plugin goal到phase上去，比如：

类似于resources:resources这种格式，说的就是resources这个plugin的resources goal（resources功能，负责处理资源文件）

process-resources resources:resources

compile compiler:compile

process-test-resources resources:testResources

test-compile compiler:testCompile

test surefire:test

package jar:jar或者war:war

install install:install

deploy deploy:deploy

site生命周期的默认绑定是：

site site:site

site-deploy site:deploy

clean生命周期的默认

clean clean:clean

到此为止，基本我们就很清楚maven的声明周期、phase和plugin的关系，同时也清楚我们运行类似mvn clean package之类的命令时，到底是在干啥了。。。。。。也知道为啥这个命令一执行，就可以实现清理、打包之类的功能更了。。。

4、maven的命令行和生命周期

比如mvn clean package

clean是指的clean生命周期中的clean phase

package是指的default生命周期中的package phase

此时就会执行clean生命周期中，在clean phase之前的所有phase和clean phase，pre clean，clean

同时会执行default生命周期中，在package phase之前的所有phase和package phase

就是这个意思

mvn clean package

clean生命周期的pre clean，clean两个phase

但是，pre clean phase默认是没有绑定任何一个plugin goal的，所以默认什么也不会干；clean phase默认是绑定了clean:clean，clean plugin的clean goal，所以就会去执行clean插件的clean goal，就会实现一个功能，就是清理target目录下的文件

process-resources resources:resources

compile compiler:compile

process-test-resources resources:testResources

test-compile compiler:testCompile

test surefire:test

package jar:jar或者war:war

install install:install

deploy deploy:deploy

mvn clean install

mvn clean deploy

mvn dependency:tree

mvn deploy:deploy-file

就是不执行任何一个生命周期的任何一个phase

直接执行指定的插件的一个goal

比如mvn dependency:tree，就是直接执行dependency这个插件的tree这个goal，这个意思就是会自动分析pom.xml里面的依赖声明，递归解析所有的依赖，然后打印出一颗依赖树

mvn deploy:depoy-file，就是直接执行deploy这个插件的deploy-file这个goal，这个意思就是说将指定目录的jar包，以指定的坐标，部署到指标的maven私服仓库里去，同时使用指定仓库id对应的server的账号和密码。

下一讲，预告，大家用过maven的人应该都知道，maven里很重要的一块配置，就是各种各样的plugin，然后每个plugin内部都有一些乱七八糟的配置，各种东西，以前看的晕晕乎乎的，但是现在在透彻理解了maven生命周期执行原理之后，你首先就知道了插件是怎么回事。。。