

Maven生命周期

2021年9月19日 22:06

maven是用来管理项目的构建的，这是他最主要的功能之一
它主要通过生命周期，来定义项目的构建顺序

Maven的三个生命周期

Clean（清理）

主要用来做清理工作，即删除上次构建中所产生的各种文件（即target文件夹下）

Default（构建）

Maven最主要的生命周期，项目的编译、测试、打包、安装部署等都在这个生命周期

Site（站点）

用来生成项目的一些报告，如测试报告、API文档等

（Maven可以根据POM中的定义信息，生成我们所需要的文档，发布到指定的Web服务器上，方便团队的相互交流）

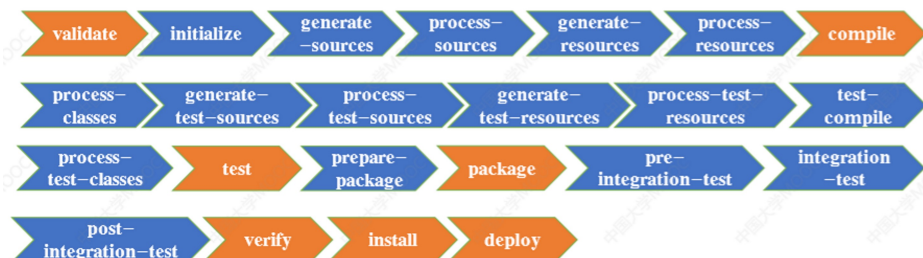
Clean生命周期阶段

1. pre-clean
在clean之前，做一些预备工作
2. **clean**
最主要的阶段，清理上一个构建所生成的文件
3. post-clean
在clean之后，做一些善后工作



★ Default生命周期

1. **validate**
验证项目中间每个目录下所需要的资源是否是可用的
2. initialize
做一些初始化的工作，如初始化一些属性、创建一些目录等
3. generate-sources
在这个阶段，可以用一些插件去生成一些额外的源代码
（如用Lombok为pojo对象、模型对象生成set/get方法）
4. process-sources
处理源代码，对源代码里的一些值进行替换等
5. generate-resources
生成资源文件，这里的资源文件指配置文件，如MyBatis所需要的xml
如果这些资源文件需要自动生成，那么就是在这个阶段生成
6. process-resources
用来处理这些资源文件，如做替换，为打包做一些准备等
7. **compile**
把源代码编译成.class文件
8. process-classes
对编译出来的.class文件做一些处理，如对字节码做一些增强处理，这些插件就在这个阶段运行
9. generate-test-sources
生成测试源代码
10. process-test-sources
处理测试源代码，如替换一些值
11. generate-test-resources
生成测试用的资源文件
12. process-test-resources
处理测试用的资源文件
13. test-compile
编译项目单元测试的代码
14. process-test-classes
对测试编译出来的.class做一些处理
15. **test**
真正的测试阶段，用来做单元测试，会使用一些单元测试的框架来进行单元测试
注意，在这个阶段，测试代码只是在运行，不会被打包，也不会被部署到服务器
16. prepare-package
打包之前的预备工作，通常是做解压，如处理包的版本
17. **package**
打包阶段，根据我们在POM文件中的定义创建jar包或war包
18. pre-integration-test
做集成测试之前的一些预处理工作，如建立相应的环境等
19. integration-test



把jar包部署到服务器上去，在测试环境中进行集成测试

20. post-integration-test

测试后的清理工作

21. **verify**

对集成测试的结果进行检查，看质量是否达标，如果质量达标进入下一个install阶段

22. **install**

把打包的项目部署到本地的Maven仓库，即用户目录的.m2文件夹

这样在本地的其他项目就可以在本地仓库中间找到对应的jar包

23. **deploy**

把jar包部署到远端的仓库中，项目组的其他成员就可以访问到对应的jar包

Site生命周期的阶段

1. pre-site

生成项目文档之前的工作

2. **site**

生成项目文档

3. post-site

生成项目文档之后的工作

4. site-deploy

把项目文档发布到指定的Web服务器上



为什么Maven要定义三个生命周期

每一个生命周期中各个阶段是有先后顺序的，**当我们执行一个阶段时，这个阶段之前的阶段都会执行**

如：我们执行Default周期中的test阶段，那么Default周期中从validate到process-test-classes都会执行

这三个生命周期之间是**相互独立**的，如我们执行Default周期的某个阶段时，不会执行Clean或Site的某个阶段

这样的设定让管理变得比较简单，要做什么事只需要执行对应生命周期的对应阶段即可，**减少了执行命令的输入**

每个生命周期中的各个阶段定义了干活的顺序，但真正干活的是在Maven中的插件，**生命周期的每一个阶段都会和插件绑定**
当我们执行某个阶段时，其实是执行该阶段所绑定的插件的功能