

Maven插件

2021年9月20日 21:21

Maven干活是依靠插件来干活的，生命周期的阶段只是定义了干活的顺序，每个阶段是要绑定对应的插件的

每个插件可以完成多个功能，插件的每一个功能成为插件的目标，即Plugin Goal

以最常用的spring-boot-maven-plugin为例

这个插件包含了多个功能/目标

- **springboot:run**
最常用，主要用来运行SpringBoot的应用程序
这个插件会在Maven的标准目录结构中找到jar文件，即在target下找到jar文件然后运行
- **springboot:repackage**
进行重打包，重新打jar包或war包，把它变成一个可执行的包
- **springboot:help**
显示springboot plugin的一些帮助信息
- **springboot:start**
用来启动SpringBoot应用程序，但是不会阻塞，允许在运行一个应用程序时再运行一个
在用springboot:run时会运行一个jar包，会被阻塞，当前目标执行完之后才会继续
- **springboot:stop**
停止用start运行起来的应用程序

这个插件其实并没有和Maven生命周期的阶段绑定，只是将其作为一个plugin引进来

plugin和Maven生命周期阶段绑定的方式

内置绑定

Maven中已经被默认绑定好了的

如：

Clean生命周期的内置绑定

阶段	内置绑定的插件目标
clean	maven-clean-plugin:clean

在Default生命周期中，根据packaging打包方式的不同来决定阶段怎样绑定

以jar包为例：

阶段	内置绑定的插件目标
process-resource	maven-resources-plugin:resources
compiler	maven-compiler-plugin:compile
process-test-resources	maven-resources-plugin:testResources
test-compile	maven-compiler-plugin:testCompile
test	maven-surefire-plugin:test
package	maven-jar-plugin:jar/maven-war-plugin:war
install	maven-install-plugin:install
deploy	maven-deploy-plugin:deploy

如果packaging的值是pom，这种情况是用在父子项目管理中的，在父项目中是不需要去打包的，只需要定义在所有子项目中所共享的一些信息

这种情况下只绑定了install和deploy

Site生命周期的内置绑定

阶段	内置绑定的插件目标
site	maven-site-plugin:site
site-deploy	maven-site-plugin:deploy

自定义绑定

如果我们需要在Maven的不同阶段去做一些我们自己定义的一些动作、功能等，就需要在plugins中自定义阶段和插件的绑定关系