# Maven插件

1. <https://www.yiibai.com/maven/maven_plugins.html>

2. <https://www.iteye.com/blog/juvenshun-232916>

## 一、什么是Maven的插件?

Maven是一个执行插件的框架，每一个任务实际上是有由插件完成的。

每个生命周期中都包含着一系列的阶段（Phase），这些阶段就相当于Maven提供的统一的接口，然后这些phase的实现是由Maven的插件来完成。

如果你理解设计模式中的模板方法（Template Method）模式，那么这会变得很好理解。

在模板方法模式中，父类定义了一些抽象方法，并且决定这些方法的调用顺序，子类只需要实现这些方法。

比如沏茶这个用例，父类定义的抽象方法有：烧开水，洗茶杯，倒水，放茶叶，并且有一个沏茶方法来顺序调用这些方法。而子类会去实现如何烧开水，如何洗茶杯，如何倒水，如何放茶叶。实例化一个子类之后，调用沏茶方法，这些步骤就会依次执行了。

同样的道理，在Maven中，生命周期定义了编译，测试，打包，部署等等阶段，而插件会去实现这些对应的阶段，如上面提到的Compiler插件实现了如何编译。

一个插件通常提供了一组目标，而目标通常是和生命周期中的某个阶段绑定在一起，一个插件目标可以绑定到多个生命周期的阶段，可使用如下语法来执行

mvn [plugin-name]:[goal-name]

## 二、插件类型：

Maven提供了以下两种类型插件：

|  |  |
| --- | --- |
| 类型 | 描述 |
| 构建插件 | 在生成过程中执行,并在pom.xml中的<build/>元素进行配置 |
| 报告插件 | 在网站生成期间执行,在pom.xml中的<reporting/>元素进行配置 |

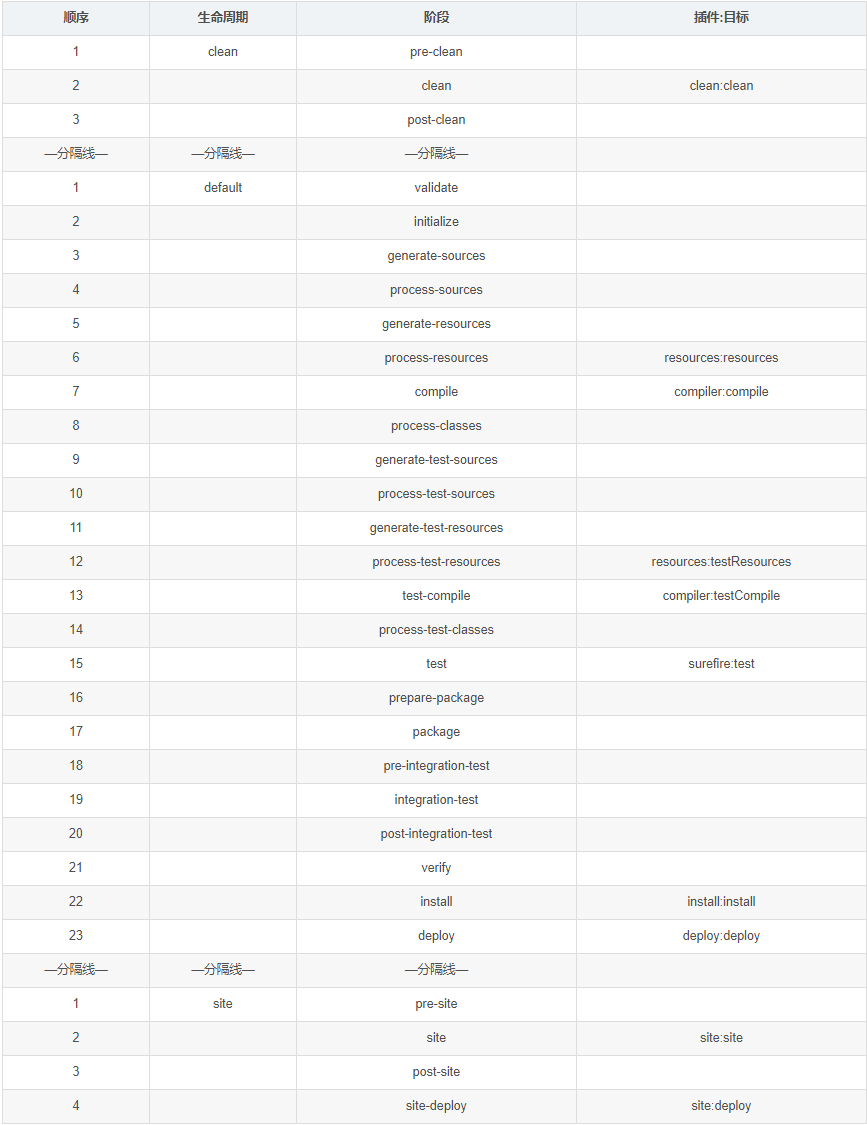
## 三、常见插件列表：

|  |  |
| --- | --- |
| 插件 | 描述 |
| clean | 编译后的清理目标,删除目标目录 |
| compile | 编译Java源文件 |
| surefire | 进行单元测试,创建测试报告 |
| jar | 从当前项目构建JAR文件 |
| war | 从当前项目构建WAR文件 |
| javadoc | 产生用于该项目的javadoc |
| antrun | 从构建的任何阶段运行一组Ant任务 |

## 四、默认生命周期阶段与插件目标的绑定关系：

Maven为实现快速构建，有一套内置的插件绑定。

三套生命周期的插件绑定如下（其实是各个生命周期阶段与插件目标的绑定）



## 五、示例：

### 1、antrun插件：

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.projectgroup</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.1</version>

<executions>

<execution>

<id>id.clean</id>

<phase>clean</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo>clean phase</echo>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</project>

【解读】

这段代码的意思是：

当进行到了Clean生命周期的clean阶段(<phase>clean</phase>)的时候，执行antrun插件(maven-antrun-plugin)的run目标(<goal>run</goal>)，而该目标所要做的任务是输出“clean phase”(<tasks><echo>clean phase</echo></tasks>)。

### 2、Tomcat插件：

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.projectgroup</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

<build>

<plugins>

<!-- 配置Tomcat插件 -->

<plugin>

<!-- https://mvnrepository.com/artifact/org.apache.tomcat/tomcat-catalina -->

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.1</version>

<configuration>

<path>/</path>

<port>8080</port>

</configuration>

</plugin>

</plugins>

</build>

</project>

配置了Tomcat插件之后，运行web项目的时候，就不用我们本地下载Tomcat服务器

### 3、compile插件：

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.companyname.projectgroup</groupId>

<artifactId>project</artifactId>

<version>1.0</version>

<build>

<plugins>

<!-- 配置编译插件 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<encoding>utf-8</encoding>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin></plugins>

</build>

</project>

4、help插件：

Maven Help插件有四个目标：

1. help:active-profiles

列出当前构建中活动的profile（项目的，用户的，全局的）

1. help:effective-pom

显示当前构建的实际POM，包含活动的profile

1. help:effective-settings

打印出项目的实际settings，包括从全局的settings和用户的settings继承的配置。

1. help:describe

描述插件的属性，它不需要在项目目录下运行。但是你必须提供你想要描述插件的groupId和artifactId。

## 六、描述一个Maven插件：

一旦你开始使用Maven，你会花很多时间去试图获得Maven插件的信息：插件如何工作？配置参数是什么？目标是什么？你会经常使用help:describe目标来获取这些信息。通过plugin参数，你可以指定想要研究哪个插件，你可以传入插件的前缀，如help就是maven-help-plugin插件的前缀，或者可以是groupId:artifact[:version]，这里的version是可选的。比如，下面的命令使用help插件的describe目标来输出Maven Help插件的信息。

C:\Dev\Repository\ProjectRepository\HelloProject>mvn help:describe -Dplugin=help

[INFO] Scanning for projects...

[INFO]

[INFO] ---------------------< com.chenming:HelloProject >----------------------

[INFO] Building HelloProject 1.0

[INFO] --------------------------------[ pom ]---------------------------------

[INFO]

[INFO] --- maven-help-plugin:3.2.0:describe (default-cli) @ HelloProject ---

[INFO] org.apache.maven.plugins:maven-help-plugin:3.2.0

Name: Apache Maven Help Plugin

Description: The Maven Help plugin provides goals aimed at helping to make

sense out of the build environment. It includes the ability to view the

effective POM and settings files, after inheritance and active profiles have

been applied, as well as a describe a particular plugin goal to give usage

information.

Group Id: org.apache.maven.plugins

Artifact Id: maven-help-plugin

Version: 3.2.0

Goal Prefix: help

This plugin has 8 goals:

help:active-profiles

Description: Displays a list of the profiles which are currently active for

this build.

help:all-profiles

Description: Displays a list of available profiles under the current

project.

Note: it will list all profiles for a project. If a profile comes up with a

status inactive then there might be a need to set profile activation

switches/property.

help:describe

Description: Displays a list of the attributes for a Maven Plugin and/or

goals (aka Mojo - Maven plain Old Java Object).

help:effective-pom

Description: Displays the effective POM as an XML for this build, with the

active profiles factored in, or a specified artifact. If verbose, a comment

is added to each XML element describing the origin of the line.

help:effective-settings

Description: Displays the calculated settings as XML for this project,

given any profile enhancement and the inheritance of the global settings

into the user-level settings.

help:evaluate

Description: Evaluates Maven expressions given by the user in an

interactive mode.

help:help

Description: Display help information on maven-help-plugin.

Call mvn help:help -Ddetail=true -Dgoal=<goal-name> to display parameter

details.

help:system

Description: Displays a list of the platform details like system properties

and environment variables.

For more information, run 'mvn help:describe [...] -Ddetail'

通过设置plugin参数来运行describe目标，输出为该插件的Maven坐标，目标前缀，和该插件的一个简要介绍。尽管这些信息非常有帮助，你通常还是需要了解更多的详情。如果你想help插件输出完整的带有参数的目标列表，只要运行带有参数detail的help:describe目标就可以了，像这样：

C:\Dev\Repository\ProjectRepository\HelloProject>mvn help:describe -Dplugin=help -Ddetail

[INFO] Scanning for projects...

[INFO]

[INFO] ---------------------< com.chenming:HelloProject >----------------------

[INFO] Building HelloProject 1.0

[INFO] --------------------------------[ pom ]---------------------------------

[INFO]

[INFO] --- maven-help-plugin:3.2.0:describe (default-cli) @ HelloProject ---

[INFO] org.apache.maven.plugins:maven-help-plugin:3.2.0

Name: Apache Maven Help Plugin

Description: The Maven Help plugin provides goals aimed at helping to make

sense out of the build environment. It includes the ability to view the

effective POM and settings files, after inheritance and active profiles have

been applied, as well as a describe a particular plugin goal to give usage

information.

Group Id: org.apache.maven.plugins

Artifact Id: maven-help-plugin

Version: 3.2.0

Goal Prefix: help

This plugin has 8 goals:

help:active-profiles

Description: Displays a list of the profiles which are currently active for

this build.

Implementation: org.apache.maven.plugins.help.ActiveProfilesMojo

Language: java

Available parameters:

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.

help:all-profiles

Description: Displays a list of available profiles under the current

project.

Note: it will list all profiles for a project. If a profile comes up with a

status inactive then there might be a need to set profile activation

switches/property.

Implementation: org.apache.maven.plugins.help.AllProfilesMojo

Language: java

Available parameters:

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.

help:describe

Description: Displays a list of the attributes for a Maven Plugin and/or

goals (aka Mojo - Maven plain Old Java Object).

Implementation: org.apache.maven.plugins.help.DescribeMojo

Language: java

Available parameters:

artifactId

User property: artifactId

The Maven Plugin artifactId to describe.

Note: Should be used with groupId parameter.

cmd

User property: cmd

A Maven command like a single goal or a single phase following the Maven

command line:

mvn [options] [<goal(s)>] [<phase(s)>]

detail (Default: false)

User property: detail

This flag specifies that a detailed (verbose) list of goal (Mojo)

information should be given.

goal

User property: goal

The goal name of a Mojo to describe within the specified Maven Plugin. If

this parameter is specified, only the corresponding goal (Mojo) will be

described, rather than the whole Plugin.

groupId

User property: groupId

The Maven Plugin groupId to describe.

Note: Should be used with artifactId parameter.

minimal (Default: false)

User property: minimal

This flag specifies that a minimal list of goal (Mojo) information should

be given.

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.

plugin

Alias: prefix

User property: plugin

The Maven Plugin to describe. This must be specified in one of three

ways:

1. plugin-prefix, i.e. 'help'

2. groupId:artifactId, i.e. 'org.apache.maven.plugins:maven-help-plugin'

3. groupId:artifactId:version, i.e.

'org.apache.maven.plugins:maven-help-plugin:2.0'

version

User property: version

The Maven Plugin version to describe.

Note: Should be used with groupId/artifactId parameters.

help:effective-pom

Description: Displays the effective POM as an XML for this build, with the

active profiles factored in, or a specified artifact. If verbose, a comment

is added to each XML element describing the origin of the line.

Implementation: org.apache.maven.plugins.help.EffectivePomMojo

Language: java

Available parameters:

artifact

User property: artifact

The artifact for which to display the effective POM.

Note: Should respect the Maven format, i.e. groupId:artifactId[:version].

The latest version of the artifact will be used when no version is

specified.

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.

verbose (Default: false)

User property: verbose

Output POM input location as comments.

help:effective-settings

Description: Displays the calculated settings as XML for this project,

given any profile enhancement and the inheritance of the global settings

into the user-level settings.

Implementation: org.apache.maven.plugins.help.EffectiveSettingsMojo

Language: java

Available parameters:

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.

showPasswords (Default: false)

User property: showPasswords

For security reasons, all passwords are hidden by default. Set this to

true to show all passwords.

help:evaluate

Description: Evaluates Maven expressions given by the user in an

interactive mode.

Implementation: org.apache.maven.plugins.help.EvaluateMojo

Language: java

Available parameters:

artifact

User property: artifact

An artifact for evaluating Maven expressions.

Note: Should respect the Maven format, i.e. groupId:artifactId[:version].

The latest version of the artifact will be used when no version is

specified.

expression

User property: expression

An expression to evaluate instead of prompting. Note that this must not

include the surrounding ${...}.

forceStdout (Default: false)

User property: forceStdout

This options gives the option to output information in cases where the

output has been suppressed by using -q (quiet option) in Maven. This is

useful if you like to use maven-help-plugin:evaluate in a script call

(for example in bash) like this:

RESULT=$(mvn help:evaluate -Dexpression=project.version -q -DforceStdout)

echo $RESULT

This will only printout the information which has been requested by

expression to stdout.

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console. This parameter will be ignored if no

expression is specified.

Note: Could be a relative path.

help:help

Description: Display help information on maven-help-plugin.

Call mvn help:help -Ddetail=true -Dgoal=<goal-name> to display parameter

details.

Implementation: org.apache.maven.plugins.help.HelpMojo

Language: java

Available parameters:

detail (Default: false)

User property: detail

If true, display all settable properties for each goal.

goal

User property: goal

The name of the goal for which to show help. If unspecified, all goals

will be displayed.

indentSize (Default: 2)

User property: indentSize

The number of spaces per indentation level, should be positive.

lineLength (Default: 80)

User property: lineLength

The maximum length of a display line, should be positive.

help:system

Description: Displays a list of the platform details like system properties

and environment variables.

Implementation: org.apache.maven.plugins.help.SystemMojo

Language: java

Available parameters:

output

User property: output

Optional parameter to write the output of this help in a given file,

instead of writing to the console.

Note: Could be a relative path.