

Maven3 实战笔记 15 编写 Maven 插件

刘岩

Email:suhuanzheng7784877@163.com

1. 自定义插件介绍

在一些情况下，Maven 现有的一些插件可能满足不了我们的一些“特殊服务”。这个时候有三种选择，第一，问问有经验的 Maven 开发者，是否有相应的。第二，从网上找找搜搜，看看有没有别的组织开发的 Maven 插件可以使用。第三，以上两条路都没走通，看来只能自己开发 Maven 的插件了，谁让咱们的需求比较“特殊”呢！“特殊服务”就得付出点代价不是~对于插件，大家可能接触最深的就是 IDE 插件了，像 Eclipse 这种 IDE 就开创了插件时代，后来的 NetBeans、IntelliJ 都是内核比较少，插件加在一起显得比较重量级。

2. 编写简单 Maven 插件

编写一个 Maven 插件其实不难，关键是编写优秀的 Maven 插件确实不容易。相对于 Eclipse Plugin 开发来说，Maven 插件开发显得简单又清晰。只需要注意两点就可以开发出一个简单的插件，第一，编写插件 pom.xml 文件；第二，编写真正做事的插件 Java 类。

咱先看 pom.xml 文件内容啊

```
<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/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.liuyan.account</groupId>
    <artifactId>MavenAccount-Plugin</artifactId>
    <version>1.0</version>
    <packaging>maven-plugin</packaging>
    <name>MavenAccount-Plugin</name>
    <url>http://maven.apache.org</url>
```

```

    <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.apache.maven</groupId>
            <artifactId>maven-plugin-api</artifactId>
            <version>3.0.3</version>
        </dependency>
    </dependencies>
</project>

```

这个 pom.xml 文件主要注意两点，一个就是<packaging>的类型，应该是 maven-plugin，还有一个就是 Maven 插件项目必须依赖一个插件 maven-plugin-api。

之后咱们再来看真正为客户端提供“特殊”Maven“服务”的类

```

package com.liuyan.account.MavenAccount_Plugin;

import org.apache.maven.plugin.AbstractMojo;
import org.apache.maven.plugin.MojoExecutionException;
import org.apache.maven.plugin.MojoFailureException;

/**
 * @goal info
 * @phase compile
 * @requiresProject false
 */
public class HelloWorldMojo extends AbstractMojo {

    /**
     * @parameter expression="${name}"
     * @required
     */
}

```

```

String name;

/**
 * @parameter expression="${age}"
 * @required
 */
int age;

/**
 * @parameter expression="${isOk}"
 * @required
 */
boolean isOk;

public void execute() throws MojoExecutionException,
MojoFailureException {
    getLog().info(this.toString());
}

public String toString() {
    return "String is : \"" + name + "\"" + "int is : \"" + age +
    "\""
        + "boolean is : \"" + isOk + "\"";
}
}

```

插件类必须继承自类 `org.apache.maven.plugin.AbstractMojo`。这里面的一些注解比较有意思，大家别当它是普通的注释信息，Maven 编译的时候不会忽略这些注释信息的，这些注释信息暗含了该插件的执行信息。

`@goal info`：表示该插件的服务目标

`@phase compile`：表示该插件的生效周期阶段

`@requiresProject false`：表示是否依托于一个项目才能运行该插件

`@parameter expression="${name}"`：表示插件参数，使用插件的时候会用到

`@required`:代表该参数不能省略

3. 使用自定义插件

使用插件之前先进行安装 install。

之后在命令行输入如下命令

```
com.liuyan.account:MavenAccount-Plugin:info -Dname=liuyan -Dage=25 -DisOk=true
```

输出如下效果

```
[INFO] --- MavenAccount-Plugin:1.0:info (default-cli) @ MavenAccount-Plugin ---  
  
[INFO] String is : "liuyan"int is : "25"boolean is : "true"  
  
[INFO] -----  
  
[INFO] BUILD SUCCESS  
  
[INFO] -----  
  
[INFO] Total time: 0.609s  
  
[INFO] Finished at: Thu Jun 23 23:17:45 CST 2011  
  
[INFO] Final Memory: 2M/15M  
  
[INFO] -----
```

插件输出了相关信息。您可以为您的服务提供更特殊的定制化“服务”。