



链滴

# SpringCloud Alibaba 微服务实战四 - 版本管理

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1575696397457>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



导读：本篇作为SpringCloud Alibaba微服务实战系列的第四篇，我们对之前的微服务框架进行优化通过Maven bom机制管理所有组件的版本。

## BOM 简介

BOM (Bill of Materials) 是由Maven提供的功能,它通过定义一整套相互兼容的jar包版本集合, 使时只需要依赖该BOM文件, 即可放心的使用需要的依赖jar包, 且无需再指定版本号。BOM的维护方负责版本升级, 并保证BOM中定义的jar包版本之间的兼容性。

## 为什么要使用BOM

使用BOM除了可以方便使用者在声明依赖的客户端时不需要指定版本号外, 最主要的原因是可以解依赖冲突, 防止你项目中出现 `NoSuchMethodError`, `ClassNotFoundException`等不可控的异常。

## 项目改造

在SpringCloud项目体系中我们约定跟SpringCloud版本相关的用主pom文件进行版本控制, 对于第三方组件或者公共模块的版本使用自定义的bom进行控制, 接下来我们一步步对原有框架进行改造。

- 建立cloud-bom模块

此模块很简单, 只有一个pom文件, 定义项目模块中所需要依赖的第三方组件和公共模块, 完整的pom文件如下:

```
<?xml version="1.0" encoding="UTF-8"?>
<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.jianzh5.cloud</groupId>
<artifactId>cloud-bom</artifactId>
<packaging>pom</packaging>
<version>1.0.0</version>

<properties>
  <mybatis-plus.version>3.1.1</mybatis-plus.version>
  <mysql.version>5.1.47</mysql.version>
  <cloud-alibaba.version>1.0.0</cloud-alibaba.version>
</properties>
<!--管理所有第三方jar包版本，SpringCloud Alibaba 版本由主Pom控制-->
<dependencyManagement>
  <dependencies>
    <!--cloud-common 版本-->
    <dependency>
      <groupId>com.jianzh5.cloud</groupId>
      <artifactId>cloud-common</artifactId>
      <version>${cloud-alibaba.version}</version>
    </dependency>
    <dependency>
      <groupId>com.jianzh5.cloud</groupId>
      <artifactId>account-feign</artifactId>
      <version>${cloud-alibaba.version}</version>
    </dependency>
    <dependency>
      <groupId>com.jianzh5.cloud</groupId>
      <artifactId>product-feign</artifactId>
      <version>${cloud-alibaba.version}</version>
    </dependency>

    <!--database-->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>${mysql.version}</version>
    </dependency>
    <dependency>
      <groupId>com.baomidou</groupId>
      <artifactId>mybatis-plus-boot-starter</artifactId>
      <version>${mybatis-plus.version}</version>
    </dependency>
  </dependencies>

</dependencyManagement>

<repositories>
  <repository>
    <id>nexus-aliyun</id>
    <name>Nexus aliyun</name>
    <url>http://maven.aliyun.com/nexus/content/groups/public</url>
  </repository>
</repositories>
</project>

```

注意，这个模块不要再定义<parent>依赖了，否则在主pom文件中引入bom后项目启动会报循环依赖的异常（死循环），具体错误如下所示：

The build could not read 1 project -> [Help 1]

The project com.jianzh5.cloud:cloud-aliaba:1.0.0 (D:\project\_jianzh5\cloud-aliaba\pom.xml) as 1 error

The dependencies of type=pom and with scope=import form a cycle: com.jianzh5.cloud:cloud-bom:1.0.0 -> com.jianzh5.cloud:cloud-bom:1.0.0 @ com.jianzh5.cloud:cloud-bom:1.0.0  
To see the full stack trace of the errors, re-run Maven with the -e switch.  
Re-run Maven using the -X switch to enable full debug logging.

For more information about the errors and possible solutions, please read the following articles:

- 在主pom文件 <dependencyManagement> </dependencyManagement> 引入bom模块的pom

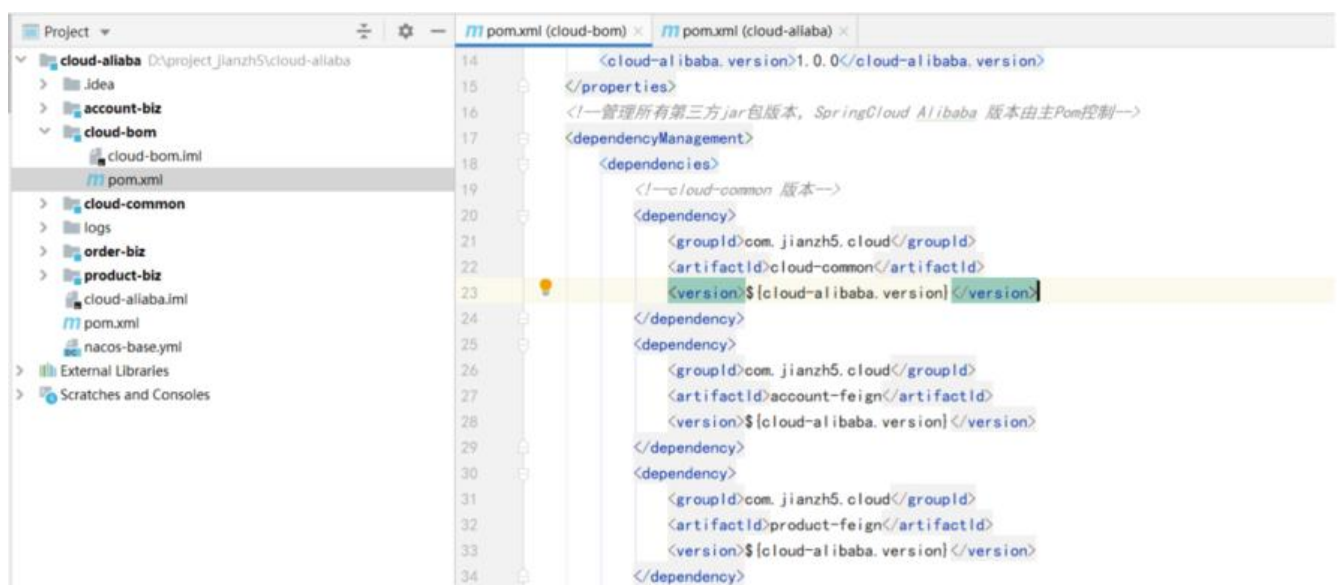
```
<!--统一版本-->
<dependency>
  <groupId>com.jianzh5.cloud</groupId>
  <artifactId>cloud-bom</artifactId>
  <version>1.0.0</version>
  <type>pom</type>
  <scope>import</scope>
</dependency>
```

此段内容需要放在dependencyManagement中第一个位置

- 去掉其他模块中引用组件的 <version> 定义，如：

```
<dependency>
  <groupId>com.jianzh5.cloud</groupId>
  <artifactId>cloud-common</artifactId>
</dependency>
```

- 改造完成后整体结构如下



- 改造完成后对原项目进行测试，保证不影响原有功能

**建议在项目初期就引入bom进行管理，后期改造会相对麻烦。**

至此我们已经完成了项目的统一版本管理，那么本期的“SpringCloud Alibaba微服务实战四 - 版本管理”篇也就该结束啦，咱们下期有缘再见！

系列文章

- [SpringCloud Alibaba微服务实战一 - 基础环境准备](#)
- [SpringCloud Alibaba微服务实战二 - 服务注册](#)
- [SpringCloud Alibaba微服务实战三 - 服务调用](#)

