

微服务中使用 Maven BOM 来管理你的服务版本

作者: [jianzh5](#)

原文链接: <https://ld246.com/article/1575447384303>

来源网站: [链滴](#)

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



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,使用时只需要依赖该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,如考虑以下的依赖场景:

项目A依赖项目B 2.1和项目C 1.2版本;
项目B 2.1依赖项目D 1.1版本;
项目C 1.2依赖项目D 1.3版本;

在该例中,项目A对于项目D的依赖就会出现冲突,按照maven dependency mediation的规则,最生效的可能是:项目A中会依赖到项目D1.1版本(就近原则,取决于路径和依赖的先后和Maven版本关系)。

在这种情况下,由于项目C依赖1.3版本的项目D,但是在运行时生效的确是1.1版本,所以在运行时很容易产生问题,如 NoSuchMethodError, ClassNotFoundException等。

如何定义BOM

BOM本质上是一个普通的POM文件,区别是对于使用方而言,生效的只有<dependencyManagement>这一个部分。只需要在<dependencyManagement>定义对外发布的客户端版本即可:

```
<?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.ydj.qd</groupId>
    <artifactId>inf-bom</artifactId>
    <version>1.0</version>
    <packaging>pom</packaging>

    <name>inf-bom</name>
    <description>第三方jar包统一管理</description>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
        <spring.version>4.3.15.RELEASE</spring.version>
    </properties>

    <dependencyManagement>
        <dependencies>

            <!-- 阿里 -->
            <!-- https://mvnrepository.com/artifact/com.alibaba/druid -->
            <dependency>
                <groupId>com.alibaba</groupId>
                <artifactId>druid</artifactId>
                <version>1.1.12</version>
            </dependency>
            <!-- https://mvnrepository.com/artifact/com.aliyun.mns/aliyun-sdk-mns -->
            <dependency>
                <groupId>com.aliyun.mns</groupId>
                <artifactId>aliyun-sdk-mns</artifactId>
                <version>1.1.8</version>
                <classifier>jar-with-dependencies</classifier>
            </dependency>
            <dependency>
                <groupId>com.alibaba</groupId>
                <artifactId>fastjson</artifactId>
                <version>1.2.29</version>
            </dependency>

            <!-- Apache -->
            <dependency>
                <groupId>org.apache.commons</groupId>
                <artifactId>commons-lang3</artifactId>
                <version>3.3.2</version>
            </dependency>
            <dependency>
                <groupId>commons-collections</groupId>
                <artifactId>commons-collections</artifactId>
                <version>3.2.2</version>
            </dependency>
            <dependency>
```

```

        <groupId>org.apache.commons</groupId>
        <artifactId>commons-collections4</artifactId>
        <version>4.1</version>
    </dependency>
    <dependency>
        <groupId>commons-beanutils</groupId>
        <artifactId>commons-beanutils</artifactId>
        <version>1.9.1</version>
    </dependency>

    <!-- 谷歌 -->
    <!-- https://mvnrepository.com/artifact/com.google.guava/guava -->
    <dependency>
        <groupId>com.google.guava</groupId>
        <artifactId>guava</artifactId>
        <version>27.0.1-jre</version>
    </dependency>
    <dependency>
        <groupId>com.google.code.gson</groupId>
        <artifactId>gson</artifactId>
        <version>2.8.5</version>
    </dependency>

    <!-- 常用工具 -->
    <dependency>
        <groupId>joda-time</groupId>
        <artifactId>joda-time</artifactId>
        <version>2.7</version>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.14.4</version>
    </dependency>

</dependencies>
</dependencyManagement>

<build>
</build>

<distributionManagement>
    <repository>
        <id>maven-releases</id>
        <name>maven-releases</name>
        <url>http://mvn.ydj.com/repository/maven-releases/</url>
    </repository>
    <snapshotRepository>
        <id>maven-snapshots</id>
        <name>maven-snapshots</name>
        <url>http://mvn.ydj.com/repository/maven-snapshots/</url>
    </snapshotRepository>

```

```
</distributionManagement>
```

```
</project>
```

项目使用方法

- 在你的项目主pom.xml文件中 `<dependencyManagement>` `</dependencyManagement>` 节点首位处加入如下:

```
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.jlxc.qd</groupId>
      <artifactId>inf-bom</artifactId>
      <version>${version}</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>

    <dependency>
      ...
    </dependency>
  </dependencies>
</dependencyManagement>
```

- 在需要使用相关JAR包的pom.xml文件中 `<dependencies>` `</dependencies>` 节点下引入如下:

```
<dependencies>
  ...
  <dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
  </dependency>

  <dependency>
    <groupId>commons-collections</groupId>
    <artifactId>commons-collections</artifactId>
  </dependency>
  ...
</dependencies>
```

如果需要使用不同于当前bom中所维护的jar包版本, 则加上 `<version>` 覆盖即可, 如:

```
<dependencies>
  ...
  <dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
  </dependency>

  <dependency>
    <groupId>commons-collections</groupId>
```

```
<artifactId>commons-collections</artifactId>  
<version>3.2.1</version>  
</dependency>  
....  
</dependencies>
```

