

Maven 依赖管理配置说明文档

本文档详细说明了项目中 `dependencyManagement` 配置的架构设计、技术实现及其使用规范。该配置旨在提供统一的依赖版本控制，保障全项目依赖的一致性、兼容性与可维护性，提升开发效率并减少潜在风险。

一、设计目标

- 统一依赖版本管理：**
在多模块项目中，通过集中化定义依赖及其版本号，避免模块间版本不一致导致的运行时冲突或不可预测的行为。
- 提升版本升级效率：**
在集中配置版本号后，仅需单点更新，即可全局生效，大幅减少人工调整和测试成本。
- 增强系统稳定性：**
通过引入成熟的 BOM (Bill of Materials) 文件（如 Spring Boot 和 Spring Cloud 官方 BOM），确保相关依赖间的兼容性，避免二义性依赖问题。
- 降低学习与维护成本：**
通过标准化依赖配置，提升配置的可读性，减少重复性定义，降低开发和维护成本。

二、依赖分类与说明

本 `dependencyManagement` 配置由以下类别组成，根据依赖的功能和领域划分：

1. 内部业务依赖（`com.tongji`）

内部依赖统一以 `com.tongji` 为 `groupId`，主要包括以下模块：

- `momento-common`：提供项目通用工具类和基础组件。
- `momento-oss-api`：封装了对象存储服务（OSS）的 API 接口。
- `momento-user-api`：定义用户相关的 RPC 接口与数据模型。
- `momento-kv-api`：提供分布式键值存储的统一接口。

版本管理： 这些模块的版本统一通过 `${revision}` 占位符定义，方便在顶层 `pom.xml` 中集中维护。

2. 日志与上下文管理组件

- `momento-spring-boot-starter-biz-operationlog`：提供业务操作日志记录的统一解决方案。
- `momento-spring-boot-starter-biz-context`：处理跨服务的业务上下文信息传递。

3. Spring 生态依赖管理

项目采用 Spring Boot 和 Spring Cloud 作为核心框架，相关 BOM 配置如下：

- `spring-boot-dependencies`：
Spring Boot 官方提供的依赖 BOM，确保 Spring Boot 核心组件间版本一致性。
- `spring-cloud-alibaba-dependencies`：
Spring Cloud Alibaba 官方 BOM，提供 Nacos、RocketMQ 等组件的版本管理。
- `spring-cloud-dependencies`：
Spring Cloud 官方 BOM，统一管理 Eureka、Feign 等微服务核心组件。

引入方式：以上 BOM 以 `<scope>import</scope>` 的形式引入，确保所有子模块均遵循统一版本策略。

4. 数据库与数据访问层

- MySQL 驱动：
 - `mysql-connector-java`：用于与 MySQL 数据库进行交互。
- MyBatis：
 - `mybatis-spring-boot-starter`：简化 MyBatis 与 Spring Boot 的集成。
- Druid 数据库连接池：
 - `druid-spring-boot-3-starter`：提供高性能数据库连接池及监控功能。

5. 权限认证与会话管理

- `sa-token-spring-boot3-starter`：支持权限认证、会话管理的轻量化框架。
- `sa-token-redis-jackson`：实现基于 Redis 的分布式会话管理，采用 Jackson 作为序列化方式。

6. 分布式配置与服务

- Nacos：
 - `nacos-config-spring-boot-starter`：实现分布式配置管理的 Nacos 客户端。
- 分布式线程上下文：
 - `transmittable-thread-local`：解决线程池中 `ThreadLocal` 变量丢失问题。

7. 工具库与增强组件

- **Google Guava**：
提供高效的集合、缓存、字符串处理等功能。
- **Hutool**：
一站式 Java 工具库，覆盖常见开发场景。
- **Apache Commons Lang3**：
提供 Java 标准库未覆盖的常用工具方法。

8. 序列化与兼容性

- Jackson:
 - `jackson-core`、`jackson-databind`：JSON 序列化和反序列化的核心组件。
 - JAXB (Java Architecture for XML Binding) :
 - `jaxb-api`、`jaxb-runtime`：支持 XML 序列化，兼容老版本框架需求。
-

9. 消息中间件与异步处理

- RocketMQ:
 - `rocketmq-spring-boot-starter`：提供 RocketMQ 与 Spring Boot 的集成支持。
 - 缓冲触发器：
 - `buffer-trigger`：适用于批量数据处理和缓冲任务触发场景。
-

10. 对象存储

- 阿里云 OSS:
 - `aliyun-sdk-oss`：封装阿里云对象存储操作的官方 SDK。
 - MinIO:
 - `minio`：兼容 AWS S3 接口的对象存储客户端。
-

三、使用规范

1. 集中版本定义：

所有依赖的版本号通过 `<properties>` 节集中管理，例如：

```
1 <properties>
2   <!-- 依赖包版本 -->
3   <spring-boot.version>3.0.2</spring-boot.version>
4   <spring-cloud-alibaba.version>2022.0.0.0</spring-cloud-
  alibaba.version>
5   <spring-cloud.version>2022.0.0</spring-cloud.version>
6   <lombok.version>1.18.30</lombok.version>
7   <jackson.version>2.16.1</jackson.version>
8   ...
9 </properties>
```

2. 子模块引用规则：

子模块中无需显式声明版本号，仅需在 `<dependencies>` 中引入依赖：

```
1 <dependency>
2   <groupId>com.tongji</groupId>
3   <artifactId>momento-common</artifactId>
4 </dependency>
```

Maven 将自动继承顶层 `dependencyManagement` 中定义的版本。

3. 版本冲突处理:

如遇版本冲突, 应优先调整 `dependencyManagement` 中的统一版本, 避免在子模块中强行覆盖。

完整父pom的dependency如下:

```
1      <!-- 统一依赖管理 -->
2      <dependencyManagement>
3          <dependencies>
4              <dependency>
5                  <groupId>com.tongji</groupId>
6                  <artifactId>momento-common</artifactId>
7                  <version>${revision}</version>
8              </dependency>
9
10             <dependency>
11                 <groupId>com.tongji</groupId>
12                 <artifactId>momento-oss-api</artifactId>
13                 <version>${revision}</version>
14             </dependency>
15
16             <dependency>
17                 <groupId>com.tongji</groupId>
18                 <artifactId>momento-user-api</artifactId>
19                 <version>${revision}</version>
20             </dependency>
21
22             <dependency>
23                 <groupId>com.tongji</groupId>
24                 <artifactId>momento-kv-api</artifactId>
25                 <version>${revision}</version>
26             </dependency>
27
28             <!-- 业务接口日志组件 -->
29             <dependency>
30                 <groupId>com.tongji</groupId>
31                 <artifactId>momento-spring-boot-starter-biz-
operationlog</artifactId>
32                 <version>${revision}</version>
33             </dependency>
34
35             <!-- Jackson 组件 -->
36             <dependency>
37                 <groupId>com.tongji</groupId>
38                 <artifactId>momento-spring-boot-starter-jackson</artifactId>
39                 <version>${revision}</version>
40             </dependency>
41
42             <!-- 上下文组件 -->
43             <dependency>
44                 <groupId>com.tongji</groupId>
45                 <artifactId>momento-spring-boot-starter-biz-context</artifactId>
46                 <version>${revision}</version>
47             </dependency>
```

```
48
49 <dependency>
50   <groupId>com.tongji</groupId>
51   <artifactId>momento-distributed-id-generator-api</artifactId>
52   <version>${revision}</version>
53 </dependency>
54
55
56 <!-- Spring Boot 官方依赖管理 -->
57 <dependency>
58   <groupId>org.springframework.boot</groupId>
59   <artifactId>spring-boot-dependencies</artifactId>
60   <version>${spring-boot.version}</version>
61   <type>pom</type>
62   <scope>import</scope>
63 </dependency>
64
65 <!-- Spring Cloud Alibaba 官方依赖管理 -->
66 <dependency>
67   <groupId>com.alibaba.cloud</groupId>
68   <artifactId>spring-cloud-alibaba-dependencies</artifactId>
69   <version>${spring-cloud-alibaba.version}</version>
70   <type>pom</type>
71   <scope>import</scope>
72 </dependency>
73
74 <!-- Spring Cloud 官方依赖管理 -->
75 <dependency>
76   <groupId>org.springframework.cloud</groupId>
77   <artifactId>spring-cloud-dependencies</artifactId>
78   <version>${spring-cloud.version}</version>
79   <type>pom</type>
80   <scope>import</scope>
81 </dependency>
82
83 <!-- 避免编写那些冗余的 Java 样板式代码，如 get、set 方法等 -->
84 <dependency>
85   <groupId>org.projectlombok</groupId>
86   <artifactId>lombok</artifactId>
87   <version>${lombok.version}</version>
88 </dependency>
89
90 <!-- Jackson -->
91 <dependency>
92   <groupId>com.fasterxml.jackson.core</groupId>
93   <artifactId>jackson-databind</artifactId>
94   <version>${jackson.version}</version>
95 </dependency>
96
97 <dependency>
98   <groupId>com.fasterxml.jackson.core</groupId>
99   <artifactId>jackson-core</artifactId>
100   <version>${jackson.version}</version>
101 </dependency>
102
103 <!-- Mybatis -->
```

```
104 <dependency>
105     <groupId>org.mybatis.spring.boot</groupId>
106     <artifactId>mybatis-spring-boot-starter</artifactId>
107     <version>${spring-boot.version}</version>
108 </dependency>
109
110 <!-- MySQL 驱动 -->
111 <dependency>
112     <groupId>mysql</groupId>
113     <artifactId>mysql-connector-java</artifactId>
114     <version>${mysql-connector-java.version}</version>
115 </dependency>
116
117 <!-- Druid 数据库连接池 -->
118 <dependency>
119     <groupId>com.alibaba</groupId>
120     <artifactId>druid-spring-boot-3-starter</artifactId>
121     <version>${druid.version}</version>
122 </dependency>
123
124 <!-- Sa-Token 权限认证 -->
125 <dependency>
126     <groupId>cn.dev33</groupId>
127     <artifactId>sa-token-spring-boot3-starter</artifactId>
128     <version>${sa-token.version}</version>
129 </dependency>
130
131 <dependency>
132     <groupId>cn.dev33</groupId>
133     <artifactId>sa-token-reactor-spring-boot3-starter</artifactId>
134     <version>${sa-token.version}</version>
135 </dependency>
136
137 <!-- Sa-Token 整合 Redis （使用 jackson 序列化方式） -->
138 <dependency>
139     <groupId>cn.dev33</groupId>
140     <artifactId>sa-token-redis-jackson</artifactId>
141     <version>${sa-token.version}</version>
142 </dependency>
143
144 <!-- 相关工具类 -->
145 <dependency>
146     <groupId>com.google.guava</groupId>
147     <artifactId>guava</artifactId>
148     <version>${guava.version}</version>
149 </dependency>
150
151 <dependency>
152     <groupId>cn.hutool</groupId>
153     <artifactId>hutool-all</artifactId>
154     <version>${hutool.version}</version>
155 </dependency>
156
157 <dependency>
158     <groupId>org.apache.commons</groupId>
159     <artifactId>commons-lang3</artifactId>
```

```
160     <version>${commons-lang3.version}</version>
161 </dependency>
162
163 <!-- 阿里云短信发送 -->
164 <dependency>
165     <groupId>com.aliyun</groupId>
166     <artifactId>dysmsapi20170525</artifactId>
167     <version>3.0.0</version>
168 </dependency>
169
170 <!-- Nacos 配置中心 -->
171 <dependency>
172     <groupId>com.alibaba.boot</groupId>
173     <artifactId>nacos-config-spring-boot-starter</artifactId>
174     <version>${nacos-config.version}</version>
175 </dependency>
176
177 <dependency>
178     <groupId>com.alibaba</groupId>
179     <artifactId>transmittable-thread-local</artifactId>
180     <version>${transmittable-thread-local.version}</version>
181 </dependency>
182
183 <!-- 对象存储 Minio -->
184 <dependency>
185     <groupId>io.minio</groupId>
186     <artifactId>minio</artifactId>
187     <version>${minio.version}</version>
188 </dependency>
189
190 <!-- 阿里云 OSS -->
191 <dependency>
192     <groupId>com.aliyun.oss</groupId>
193     <artifactId>aliyun-sdk-oss</artifactId>
194     <version>${aliyun-sdk-oss.version}</version>
195 </dependency>
196
197 <dependency>
198     <groupId>javax.xml.bind</groupId>
199     <artifactId>jaxb-api</artifactId>
200     <version>${jaxb-api.version}</version>
201 </dependency>
202 <dependency>
203     <groupId>javax.activation</groupId>
204     <artifactId>activation</artifactId>
205     <version>${activation.version}</version>
206 </dependency>
207
208 <!-- no more than 2.3.3-->
209 <dependency>
210     <groupId>org.glassfish.jaxb</groupId>
211     <artifactId>jaxb-runtime</artifactId>
212     <version>${jaxb-runtime.version}</version>
213 </dependency>
214
215 <!-- Feign 表单提交 -->
```

```
216     <dependency>
217         <groupId>io.github.openfeign.form</groupId>
218         <artifactId>feign-form</artifactId>
219         <version>${feign-form.version}</version>
220     </dependency>
221
222     <!-- Caffeine 本地缓存 -->
223     <dependency>
224         <groupId>com.github.ben-manes.caffeine</groupId>
225         <artifactId>caffeine</artifactId>
226         <version>${caffeine.version}</version>
227     </dependency>
228
229     <!-- Rocket MQ -->
230     <dependency>
231         <groupId>org.apache.rocketmq</groupId>
232         <artifactId>rocketmq-spring-boot-starter</artifactId>
233         <version>${rocketmq.version}</version>
234     </dependency>
235
236     <!-- 快手 Buffer Trigger -->
237     <dependency>
238         <groupId>com.github.phantomthief</groupId>
239         <artifactId>buffer-trigger</artifactId>
240         <version>${buffertrigger.version}</version>
241     </dependency>
242
243 </dependencies>
244 </dependencyManagement>
```


