

二方库依赖的一些规范，出自Java开发手册（嵩山版）六、工程结构（二）二方库依赖，（阿里巴巴编码规范）

二方库：公司内部发布到中央仓库。可供公司内部其它应用依赖的库（jar包）

（二）二方库依赖

1. 【强制】定义GAV遵从以下规则：

1) **G**roupID格式：com.{公司/BU}.业务线.[子业务线]，最多4级。

说明：{公司/BU} 例如：alibaba/taobao/tmall/aliexpress等BU一级；子业务线可选。

正例：com.taobao.jstorm 或 com.alibaba.dubbo.register

2) **A**rtifactID格式：产品线名-模块名。语义不重复不遗漏，先到中央仓库去查证一下。

正例：dubbo-client / fastjson-api / jstorm-tool

3) **V**ersion：详细规定参考下方。

2. 【强制】二方库版本号命名方式：主版本号.次版本号.修订号

1) **主版本号**：产品方向改变，或者大规模API不兼容，或者架构不兼容升级。

2) **次版本号**：保持相对兼容性，增加主要功能特性，影响范围极小的API不兼容修改。

3) **修订号**：保持完全兼容性，修复BUG、新增次要功能特性等。

说明：注意起始版本号必须为：1.0.0，而不是0.0.1 正式发布的类库必须先去中央仓库进行查证，使版本号有延续性，正式版本号不允许覆盖升级。如当前版本：1.3.3，那么下一个合理的版本号：1.3.4 或 1.4.0 或 2.0.0

3. 【强制】线上应用不要依赖SNAPSHOT版本（安全包除外）。

说明：不依赖SNAPSHOT版本是保证应用发布的幂等性。另外，也可以加快编译时的打包构建。

4. 【强制】二方库的新增或升级，保持除功能点之外的其它jar包仲裁结果不变。如果有改变，必须明确评估和验证，建议进行 `dependency:resolve` 前后信息比对，如果仲裁结果完全不一致，那么通过 `dependency:tree` 命令，找出差异点，进行 `<excludes>` 排除jar包。

5. 【强制】二方库里可以定义枚举类型，参数可以使用枚举类型，但是接口返回值不允许使用枚举类型或者包含枚举类型的POJO对象。

6. 【强制】依赖于一个二方库群时，必须定义一个统一的版本变量，避免版本号不一致。

说明：依赖springframework-core,-context,-beans，它们都是同一个版本，可以定义一个变量来保存版本：\${spring.version}，定义依赖的时候，引用该版本。

7. 【强制】禁止在子项目的pom依赖中出现相同的GroupId，相同的ArtifactId，但是不同的Version。

说明：在本地调试时会使用各子项目指定的版本号，但是合并成一个war，只能有一个版本号出现在最后的lib目录中。可能出现线下调试是正确的，发布到线上却出故障的问题。

8. 【推荐】所有pom文件中的依赖声明放在 `<dependencies>` 语句块中，所有版本仲裁放在 `<dependencyManagement>` 语句块中。

说明：`<dependencyManagement>` 里只是声明版本，并不实现引入，因此子项目需要显式的声明依赖，version和scope都读取自父pom。而 `<dependencies>` 所有声明在主pom的 `<dependencies>` 里的依赖都会自动引入，并默认被所有的子项目继承。

9. 【推荐】二方库不要有配置项，最低限度不要再增加配置项。

10. 【参考】为避免应用二方库的依赖冲突问题，二方库发布者应当遵循以下原则：

1) 精简可控原则。移除一切不必要的API和依赖，只包含 Service API、必要的领域模型对象、Utils类、常量、枚举等。如果依赖其它二方库，尽量是provided引入，让二方库使用者去依赖具体版本号；无log具体实现，只依赖日志框架。

2) 稳定可追溯原则。每个版本的变化应该被记录，二方库由谁维护，源码在哪里，都需要能方便查到。除非用户主动升级版本，否则公共二方库的行为不应该发生变化。