配置库标识规范

**2021年9月**

目录

[项目环境类 3](#_Toc82890272)

[编译器 3](#_Toc82890273)

[框架及相关技术 3](#_Toc82890274)

[服务器 3](#_Toc82890275)

[文件版本命名规范及内容格式规范 3](#_Toc82890276)

[版本号概要说明 3](#_Toc82890277)

[版本号制定修改规则 4](#_Toc82890278)

[主版本号(1) 4](#_Toc82890279)

[子版本号(1) 4](#_Toc82890280)

[阶段版本号(1) 4](#_Toc82890281)

[日期版本号(051021) 4](#_Toc82890282)

[文件命名规范 4](#_Toc82890283)

[文件内容格式规范 4](#_Toc82890284)

[目录结构 5](#_Toc82890285)

[前端目录结构 5](#_Toc82890286)

[后端目录结构 6](#_Toc82890287)

[Maven约定目录格式 6](#_Toc82890288)

[Common通用模块目录 6](#_Toc82890289)

[脚本目录规范 7](#_Toc82890290)

# 项目环境类

## 编译器

前端：Visual Studio Code

后端：IDEA

## 框架及相关技术

前端：Vue＋ElementUI

后端：Spring Boot＋Mybatis

JDK版本：1.8（8.0）

## 服务器

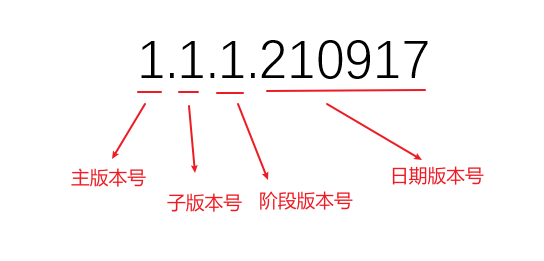
应用服务器：Tomcat

数据库：Mysql

# 文件版本命名规范及内容格式规范

## 版本号概要说明

软件版本号由四部分组成，第一个1为主版本号，第二个1为子版本号，第三个1为阶段版本号，第四部分为日期版本号。具体例子如下图所示：



## 版本号制定修改规则

### 主版本号(1)

当功能模块有较大的变动，比如增加多个模块或者整体架构发生变化。此版本号由项目决定是否修改。

### 子版本号(1)

当功能有一定的增加或变化，比如增加了对权限控制、增加自定义视图等功能。此版本号由项目决定是否修改。

### 阶段版本号(1)

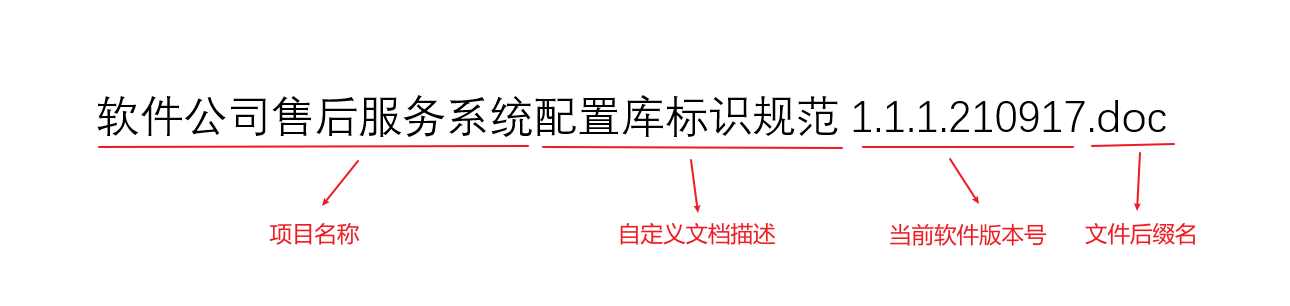
一般是 Bug 修复或是一些小的变动，要经常发布修订版，时间间隔不限，修复一个严重的bug即可发布一个修订版。此版本号由项目经理决定是否修改。

### 日期版本号(051021)

用于记录修改项目的当前日期，每天对项目的修改都需要更改日期版本号。此版本号由开发人员决定是否修改。

## 文件命名规范

文件命名规范例子如下图所示：



文件名称由四部分组成：第一部分为项目名称，第二部分为文件的描述，第三部分为当前软件的版本号，第四部分为文件后缀，例如：项目外包平台测试报告1.1.1.210917.doc，此文件为项目外包平台的测试报告文档，版本号为：1.1.1.210917。

## 文件内容格式规范

标题居中，字体宋体小三。正文为宋体五号。一级标题为小四加粗且居左宋体，二级标题为五号加粗且居左宋体。

表格的标题在表格上且为五号加粗宋体。

# 目录结构

## 前端目录结构

参考前端工程规范要求





## 后端目录结构

### Maven约定目录格式

源码目录指：{项目目录}/src/main/

打包定义目录：{项目目录}/target

代码目录：src/main/java

资源目录：src/main/resources

/db：数据库脚本归档

/data：内部依赖数据归档

/mybatis: mybatis的mapper对应xml文件

文档目录：src/main/docs

配置目录：src/main/config

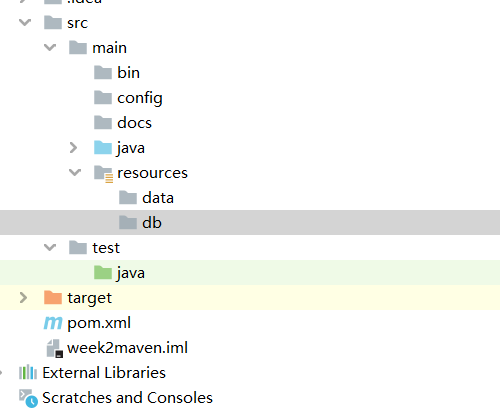
脚本目录：src/main/bin

run-manage.sh 运行管理脚本（通过参数start, stop, status, help info控制程序运行）

sh：服务组件启动脚本

//工程下还应有 .gitignore文件以及README.md文件

示例：



### Common通用模块目录

#### src 源码目录

|-- common 各个项目的通用类库

|-- |--- annotation 通用注解，比如权限，登录等等

|-- |--- constant 通用常量，比如 ResponseCodeConst

|-- |--- entity 全局的 javabean，比如 BaseEntity 等

|-- |--- exception 全局异常，如 BusinessException

|-- |--- utils 全局工具类

#### 业务模块目录

src 源码目录

|-- common 各个项目的通用类库

|-- |---constant 模块公共常量

|-- javabean 系统各种实体信息BO、VO、DTO、Entity

|-- |---dto 前端通过rpc传输的参数

|-- |---vo 返回给前端视图的参数

|-- |---entity 数据表实体映射

|-- |---bo 业务实体，可以组合多个数据实体信息在业务层使用

|-- config 项目的配置信息

|-- handler 模块处理器

|-- interceptor 模块拦截器

|-- listener 模块监听器

|-- controller 控制层接口

|-- service 业务逻辑层

|-- |--- impl 业务具体实现

|-- mapper orm层

|-- third 三方服务，比如redis, oss，微信sdk等等

|-- util 模块工具类

|-- Application.java 启动类

### 脚本目录规范

每个项目的数据库脚本维护，应该遵循一个版本的概念。在设计时，规定脚本版本号，以及升级版本号的递增规则，脚本文件命名等。

在项目工程目录下，或resource目录下维护一个 /db 目录，该目录维护项目工程在数据库设计初期产出的完整ddl文件。如果项目按版本划分，则在 /db 目录下新增对应版本号的文件夹，然后版本升级而导致脚本升级的文件则存在对应的版本号的目录里。

如 /db/version1.1.0 存放数据库设计初期所有表结构完整的DDL，后续代码升级，版本升级到version1.1.1，则在/db目录下新增/version1.1.1目录并存放此版本对数据库进行调整的增量sql文件。