

赣州之游科技有限公司

Java 项目开发规范

编写：王振宇

联系方式：13672297775

网址：<http://www.zywork.top>

时间：2017 年 08 月 10 日

为了团队开发过程中代码的可读性，可维护性，优雅性，降低代码耦合度等，提升团队开发的效率，降低团队沟通成本，特此规定赣州之游科技有限公司内部 Java 项目开发规范，此开发规范是基于阿里巴巴 Java 开发手册 v1.2.0-1 版本，在阿里巴巴 Java 开发手册的基础上做适当的补充，阿里巴巴 Java 开发手册中已说明的，不在此文件中重复提出。

一、项目结构

1) 包定义

公司域名为 `www.zywork.top`，所有公司内部产品、自营产品，`package` 都以 `top.zywork` 为前缀。非公司内部的产品，除非客户有要求，也以 `top.zywork` 为前缀。

2) MVC 架构

严格遵照 MVC 设计模式，分成架构，各个层次的包如下定义：

控制器层：`top.zywork.controller`，以 `Controller` 为后缀

服务层接口：`top.zywork.service`，以 `Service` 为后缀

服务层实现类：`top.zywork.service.impl`，以 `ServiceImpl` 为后缀

通用业务实现类：`top.zywork.manager`，以 `Manager` 为后缀

DAO 层接口：`top.zywork.dao`，以 `DAO` 为后缀或 `Mapper` 为后缀（使用 `MyBatis`）

DAO 层实现类：`top.zywork.dao.impl`，以 `DAOImpl` 为后缀或无实现类（使用 `MyBatis`）

数据对象类：`top.zywork.do`，以 `DO` 为后缀

数据传输对象类：`top.zywork.dto`，以 `DTO` 为后缀

值对象类：`top.zywork.vo`，以 `VO` 为后缀

通用工具类：`top.zywork.common`，以 `Utils` 为后缀

过滤器类：`top.zywork.filter`，以 `Filter` 为后缀

监听器类：`top.zywork.listener`，以 `Listener` 为后缀

网络操作类：`top.zywork.net`，服务端以 `Server` 为后缀，客户端以 `Client` 为后缀

3) 目录结构

文档目录：项目根目录中必须增加 `documents` 目录，用于存储项目整个开发过程中所涉及到的所有文档，包括数据库文档，需求文档，详细设计文档，用户手册等。

框架配置文件目录：如果是 `WEB` 项目，则在 `WEB-INF` 目录下创建 `config` 目录。

映射文件目录：`Hibernate` 的 `hbm` 映射文件与 `MyBatis` 的 `Mapper` 文件都存放到 `DO` 对象所在的包中。

JSP 页面目录：在 WEB-INF 目录下创建 views 目录，根据不同的模块在 views 目录下创建相应的目录。

静态文件目录：在 WebContent(Eclipse)/WebRoot(MyEclipse)/Web(IntelliJ IDEA)根目录中创建 public 目录，public 目录中 css 存放 CSS 样式文件，main.css 为主样式文件；images 存放图片文件；js 存放 JavaScript 脚本，main.js 为主 JS 脚本文件；fonts 存放图标字体文件；plugins 存放 JS 插件，一个插件对应一个目录，插件目录中包括 CSS 及 JS 文件。public 目录中 uploads 目录用于存放上传文件。

4) 访问 URL

使用 Restful API 接口的形式，任何一个 JSP 页面都不能直接以 .jsp 后缀的方式访问。绝大部分 URL 都要使用 Restful API 接口形式，少部分可以传递查询字符串，视具体情况而定。

二、文档要求

1) 数据库文档

数据库文档必须以 .sql 结尾，包含创建数据库，建表，建立约束，初始化数据的所有 SQL 语句。并且调整好建表的顺序，所有的名称避免出现关键字，任何名称不要用 ` 号括起来。

2) 需求文档

需求文档随时要保持到最新状态，如果有需求的变更，要及时更新需求文档。

3) 详细设计文档

详细的实现需要体现在文档上，并及时更新。

4) 数据库表字段设计文档

当数据库有更新时，表字段设计文档也要及时更新，并备注。

三、日志记录

1) 什么时候记录日志

当出现异常时，当重要操作时，当修改或删除数据时。

2) 日志周期

日志以天为时间间隔保存，一天对应一个日志文件。

四、项目管理

1) 项目管理工具

项目使用 Maven 或 Gradle 进行管理，所有第三方 JAR 库都使用最新稳定版本。使用过程中注意排除 JAR 库文件的版本冲突。

五、协同开发

1) 代码更新 (Update/PULL)

每天第一件事更新代码，提交代码前更新代码

2) 代码提交 (Commit/PUSH)

每次提交前先更新代码，提交代码必须写明提交信息，提交信息禁止出现：修改 bug，修改功能等类似的简单描述，一定要具体说明做了哪些事情，每一件事情按序号标出。

代码提交前做周全的单元测试，确定没问题才能提交。

及时提交确认无误可运行的代码，不要等到所有代码都写完了再提交。

3) .gitignore 文件

如果使用 Git，则必须增加.gitignore 文件，把不需要添加到版本控制的目录排除。

4) 哪些文件需要提交

与项目源代码无关的文件不要提交到版本控制仓库中，能提交的代码通常限定于 Java 源代码，JSP 页面，CSS 样式，JS 脚本，文档等。编译后生成的 class 等文件一律禁止提交。项目更新下来后，由 IDE 生成或修改的文件，视情况而提交。

六、代码编写

1) 泛型

需要用到泛型的地方，则使用泛型，并灵活运用 extends 和 super 关键字。

2) 序列化

需要序列化的类，必须增加序列化版本号，由 IDE 生成。

3) 代码警告

杜绝警告，如果警告无法通过代码优化清除掉，则使用压制警告(注意压制警告的有效位置)。

4) 单元测试

重要方法必须要做单元测试，使用 JUnit 或 SpringJUnit 做测试，测试类所在的包名称与源代码的包名称一致，测试类以 Test 为后缀。

5) JSP 静态文件的引入

JSP 页面中，样式的引入在<head>标签内部，JavaScript 文件的引入在页面的尾部，使用内联

样式和外部 JS，少部分情况可以直接使用内部样式和内部 JS。

6) JSP、静态文件命名

JSP 页面的名称、静态文件的名称，如果是多个单词，则每个单词用_连接。

7) CSS 选择器

CSS 样式文件中的类或 id，如果是多个单词，则每个单词用-连接。

8) HTML 标签 id

页面中 HTML 标签的 id，如果是多个单词，则每个单词用-连接。

9) 常用的枚举

表示 status 状态的枚举，表示性别的枚举，表示是与否的枚举等。