# 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：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，不在此开发规范中重复提出。

## 一、项目结构

### 1）包定义

公司域名为 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

严格遵循 MVC 设计模式，分成架构，各个层次的包如下定义：

控制器层类：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.dos（为了不与关键字 do 冲突，所以加 s 变成 dos），以 DO 为后缀，通过 DAO 层向上传输数据源对象

数据传输对象类：top.zywork.dto，以 DTO 为后缀，service 和 manager 向外传输的对象

值对象类：top.zywork.vo，以 VO 为后缀，向视图层传输的对象

数据查询对象：top.zywork.query，以 Query 为后缀，各层接收上层的查询请求

通用工具类：top.zywork.common，以 Utils 为后缀

常量类：top.zywork.constant，以 Constants 为后缀

枚举类：top.zywork.enums，以 Enum 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

异常类：top.zywork.exception，以 Exception 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涉及的所有文档，包括需求文档，数据库文档，详细设计文档，用户手册等。

框架配置文件目录：在 src 根目录中创建 config 目录用于存储所有框架相关的配置文件。

映射文件目录：在 src 根目录中创建 mapping 目录用于存储 Hibernate 的 hbm 映射文件，创建 mapper 目录用于存储 MyBatis 的 Mapper 文件。

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 static 目录，static 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。

上传文件目录：在 WebContent(Eclipse)/WebRoot(MyEclipse)/Web(Intellij IDEA)根目录中创建 uploads 目录用于存放上传文件，可以根据上传文件的类型在 uploads 目录中创建子目录，如 images 用于存储上传的图片文件，videos 用于存储上传的视频文件，docs 用于存储上传的 Word，PPT，Excel，TXT 等文档文件。

**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，创建表，创建约束，初始化表数据的所有 SQL 语句。并且调整好建表的顺序，所有的名称避免出现数据库关键字，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、异常与日志记录

### 1）什么时候记录日志

当出现异常时，当重要操作时如修改或删除数据，登录操作，权限分配等。

出现异常保存 error 级别的日志，操作记录保存 info 级别的日志。

### 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，日志文件的最短保留时间设置为 15 天。

### 3）自定义异常

AppException：应用异常类

DAOException：数据库异常类，继承自 AppException

ServiceException：服务异常类的父类，继承自 AppException

### 4）异常处理机制

由于使用了框架的项目，异常被封装成 RuntimeException 的子类重新抛出，故可以不捕捉任何异常，最终交由框架统一处理异常信息，但是为了更好地做异常日志记录，在每一层中加上 try…catch…语句块。

未使用框架的项目：DAO 层捕捉异常，抛出 DAOException，Service 层捕捉 DAOException 并重新抛出 ServiceException，控制器层捕捉 ServiceException 并转发到统一的错误处理页；

使用 SSH 框架的项目：DAO 层捕捉 RuntimeException，重新抛出 DAOException，Service 层捕捉 DAOException 并重新抛出 ServiceException，控制器层捕捉 ServiceException 并抛出，最终由框架统一处理异常；

使用 SSM 或 SpringBoot 的项目：Service 层捕捉 RuntimeException，重新抛出 ServiceException，控制器层捕捉 ServiceException 并抛出，最终由框架统一处理异常。

## 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 库的版本冲突。

## 五、协同开发

### 1）代码更新（Update/Pull）

每天第一件事情更新代码，提交代码前也更新代码，如果出现代码冲突，则考虑清楚如何合并代码。

**2）代码提交（Commit/Push）**

每次提交前先更新代码，如果出现代码冲突，则考虑清楚如何合并代码。

提交代码必须写明提交信息，提交信息禁止出现：更新，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，每一个序号的内容结束必须换行。

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 生成或修改的配置文件，视具体情况而提交。

## 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**

杜绝出现代码警告，如果警告无法通过代码优化清除掉，则使用压制警告，并且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 Spring JUnit 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，测试方法以 test 为前缀，测试的方法名为后缀。

**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，使用外部样式和外部 JS，少部分情况可以直接使用内部样式和内部 JS。外部静态文件的引入不使用

相对路径的方式。

**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）枚举**

表示状态的枚举，表示性别的枚举，表示是与否的枚举等，能用枚举则不用常量。

**10）Hibernate hbm 映射文件命名**

Hibernate hbm 映射文件以 DO 的简单类名开头，以.hbm.xml 结尾。

**11）MyBatis mapper 映射文件命名**

MyBatis mapper 映射文件以 DAO 的简单类名开头，以.xml 结尾。

## 七、第三方库/类使用

**1）字符串处理**

使用 commons-lang 核心包的 StringUtils 类

**2）Java Bean 拷贝**

使用 Apache Dozer 库

**3）JSON 处理**

使用阿里巴巴 fastjson 库

**4）文件处理**

使用 commons-io 包的 FileUtils 类

**5）数据源与数据连接池**

使用阿里巴巴的 druid 库

**6）Office 文档处理**

使用 Apache POI 库

**7）HTTP 网络操作**

使用 Apache HttpClient 库

**8）NIO/Socket**

使用 JBoss 提供的 Netty 或 Apache 的 Mina

**9）短信接口**

推荐使用阿里云短信接口