**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。

1. **项目结构**
2. **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。

1. **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

1. **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：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。

1. **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。

1. **文档要求**
2. **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。

1. **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。

1. **详细设计文档**

详细的实现需要体现在文档上，并及时更新。

1. **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。

1. **日志记录**
2. **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。

1. **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。

1. **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。

1. **协同开发**
2. **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。

1. **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。

1. **.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。

1. **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。

1. **代码编写**
2. **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。

1. **序列化**

需要序列化的类，必须增加序列化版本号，由IDE生成。

1. **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。

1. **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。

1. **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。

1. **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。

1. **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。

1. **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。

1. **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。