**项目团队开发规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 修改人 | 修改日期 | 修改内容 |
| 1 | 李志鹏 | 2018.06.07 | 创建 |
|  |  |  |  |

# 1 **引言**

## 编写目的

统一项目开发规范、使用的工具、环境配置。提高项目开发的效率、项目团队整体水平的提升。

## 适用读者

项目所有成员

## 编写背景

根据项目的开发情况，组件稳定、高效的项目团队。

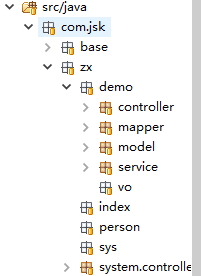
# 2详细规范

**2.1使用工具/字符集**

|  |  |  |
| --- | --- | --- |
| **开发工具/字符集** | **版本** | **说明** |
| JDK | 1.7 |  |
| ECLIPSE | 4.3 |  |
| TOMCAT | 7 |  |
| MYSQL | 5 |  |
| SVN | 3.3.1 | 服务器版本 |
| 编码格式 | UTF-8 |  |

**2.2 目录结构**

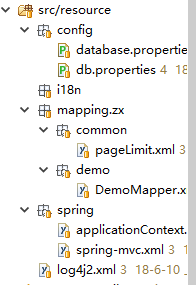
### 2.2.1 后台Java代码



demo、index、person、sys、system是对应的模块包名，分别是例子、首页、我的、系统模块。

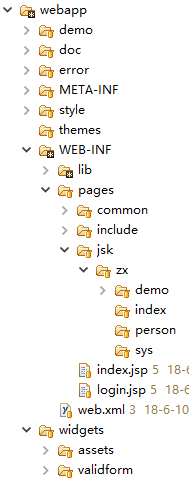
如果需要新建功能层面的包，放在zx目录下，每个包下包含controller、mapper、model、service、vo包。

### 2.2.2 配置文件

****

配置文件分为config、i18n、mapping、spring包。config是功能配置，目前是数据库配置，i18n是国际化，mapping是mybatis配置文件，spring是跟spring和springmvc相关的配置文件，log4j2是日志配置文件。

### 2.2.3 前台页面

****

该目录包含demo、doc、error、style、themes、web-inf、widgets子目录。demo是存放写的例子功能，比如针对某个小工具的介绍等；doc是存放项目相关文档，有数据库设计文档、开发规范、项目配置等；error是错误页面；style是跟项目样式相关，里面是图片、css、js等；themes是主体，以后如果项目需要多套样式可以把样式放这里面；widgets是小工具，比如现在存放了前台验证框架等；WEB-INF目录下存放项目所需要的包和web.xml，pages是项目所有页面，common功能页面、include也是功能页面、jsk是具体页面，jsk里面每个目录对应一个功能模块，跟后台保持一致。

## **2.3 命名规范**

统一使用英文单词命名，确实无法使用单词的，使用汉语拼音首字母并且要有注释。

### 2.3.1 Java

包名：如果有多个单词全部小写，不能使用驼峰命名规则命名

类名：采用驼峰命名规则

controller：model+controller，参考demo

service：model+service，参考demo。接口以大写I开头

mapper：model+mapper，参考demo

变量：变量名称尽量避免使用下划线连接两个单词

常量：名称都是大写，使用下划线连接两个单词

### 2.3.2 页面名称

列表：model\_list，参考demo

详细：model\_view，参考demo

维护：model\_form，参考demo

### 2.3.3 数据库

表名：jsk\_模块名称\_功能名称

其他规范

* + 数据表名必需大写，字段名必需大写
  + 视图必需以V\_开头
  + 数据表外键约束以FK\_表名1\_表名2
  + 数据表主键定义以PK\_表名命名
  + 触发器命名以TR\_开头
  + 存储过程命名以P\_开头
  + 函数命名以FUNC\_开头

### 2.3.4 方法

|  |  |
| --- | --- |
| **类型** | **前缀** |
| 新增 | save |
| 删除 | delete |
| 修改 | update |
| 查询 | findBy |
| 详细 | detail |

**2.4 工具格式设置**

### 2.4.1注释格式

找到项目zxWeb\webapp\doc\standard\codetemplates.xml文件，导入到eclipse里，提前把文件里的名字改下。

每段逻辑都要有注释，逻辑规则和实现思路都要写清楚，方便别人理解和错误追查，也方便自己后续看代码。

每次提交SVN都要写注释。

### 2.4.2 代码格式

#### 2.4.2.1 缩进

代码缩进4个空格

#### 2.4.2.2 空行

不要出现连续多个空行

#### 2.4.2.3 列数

设置工具每行显示字符120

#### 2.4.2.4 行数

每个方法的行数控制在30-60行之间。如果太多建议拆分成两个方法。

## 2.5 注意点

### 2.5.1 格式化代码

写完一个功能后或往SVN上提交代码前一定要格式化一下代码(快捷键CTRL+SHIFT+F)

### 2.5.2 去除多余的引用包

提交SVN 前后太类要去除多余的引用包(快捷键CTRL+SHIFT+O)

### 2.5.3 为空验证

在使用对象打点调用方法之前一定要判断是否为空

### 2.5.4 日志打印

不要使用system.out.println()打印错误信息，要使用logger输出到错误日志里。

### 2.5.5 类型转换

父类转成子类之前要使用instanceof判断下

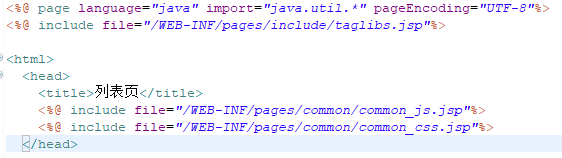
### 2.5.6 系统符号

操作文件拼接地址时要使用File.separate

获取换行符要使用System.getProperty(“line.separator”)

## 2.6 页面规范

### 2.6.1 文件引入



### 2.6.2 js位置

页面js建议都写在body结束标签下面，第一页面美观，第二提高页面加载速度。

## 2.7 系统工具

关于时间、下载、文件获取后缀名、字符串格式化等操作可以使用com.mltt.core.util包下的类里的静态方法

# 项目基础功能开发

## 3.1 controller

每个control要继承BaseEntityController，需要进行登录验证的controll路径以secure开头，BaseEntityController类里已经写好了save、delete、deleteById、create、edit、find、view、findList、findPage方法，其中findPage是分页方法，findList是不分页的。具体参考用户管理的controller。

## 3.2 service

每个service要继承IEntityService，父类已经写好了controller一样的方法。

## 3.3 model

每个实体类要继承BaseEntity类，写一个构造方法，设置返回页面，如：

public JskSysUser() {

// 如果想要baseController方法需要设置返回页面

setListPage("/jsk/zx/system/user/user\_list");

setFormPage("/jsk/zx/system/user/user\_form");

}

写一个返回表名的方法，如：

public String getTableName() {

return "JSK\_SYS\_USER";

}

## 3.4 mapper

每个mapper要继承IBaseEntityMapper类