# 开发规范:

## WEB:

#### 接口&文档

1. 定义接口之前前后台人员均熟悉业务包括但不限定于 demo、数据库设计
2. 接口需要前后端人员一起参与定义，且后台人员落实
3. 在开发之前先定义接口，包括**路径、输入参数、输出参数**
4. 当前版本增加的部分采用+且使用**粗体**， 删减的部分使用-且使用***粗斜体***，之前版本增加部分采用+（非粗体），删减的部分使用-且使用*斜体*（非粗体）
5. 接口变更需要通知相关人员

#### JSP&JavaScript&CSS

1. **命名：**
2. JS中**Dictionary.js**中记录所有前端代码中的**枚举**
3. **常量**在**Constants.js**中进行定义
4. 表格页面: XXXX-grid.jsp
5. 表格页面获取数据: XXXX-grid-load-data.html
6. 添加、编辑页面: XXXX-edit.jsp
7. 页面异步加载数据: XXXX-load-data.html
8. 页面异步加载非默认字典数据: XXXX-list-load-data.html
9. **格式：**
10. 所有人员的使用相同的格式化代码配置，宽度不超过120
11. 只有一行的代码块不能缺少{}
12. 行必须以分号；结尾
13. 数组最后一个值不能跟逗号,
14. 采用自动化格式化
15. 方法、变量命名采用**驼峰命名法、【类名大写】**，命名遵循**见名知义**原则

例如 方法名：queryPersonById、queryPersonList等

1. 代码**适当添加注释**
2. 代码块使用频率达到2次以上，需**提取成方法**
3. 通用方法可考虑封装为util类放在util包下，例如导出文件等
4. **路径：**
5. **Controller**路径采用**两级路径，短横线分隔**，一个短横线表示一个**目录**，所有前端后端代码结构目录一致
6. 所有的路径均采用**绝对路径**包括css,image，即所有的路径前需要添加contextPath
7. **提示信息:**

            前端提示信息均采用后台输出的信息，不要自定义提示信息

1. **第三方JavaScript:**

严禁开发人员任意引入第三方JavaScript，如果有需要请项目技术负责人确认

1. **测试**

            页面需要使用JSON测试数据进行单元测试

#### Java

1. **表结构：**
2. 每个的大文本建立单独的表，与基本表建立一对一关系
3. 所有的附件均存储到附件表BASE\_ATTACHMENT
4. 多对多关系以REL开头,SEC表除外
5. 所有的表结构，**初始数据变动需要同步到脚本中**，具体由项目组内一个单独的人员负责更新脚本，脚本需要使用脚本测试库进行测试
6. 基本表**伪删除**：使用两个字段DICT\_DELETED\_NAME,DICT\_DELETE\_VALUE
7. **格式：**
8. 所有人员的使用相同的格式化代码配置，宽度不超过120
9. 只有一行的代码块不能缺少{}
10. **变量命名:**
11. 后端代码中的枚举使用枚举模板类定义
12. 常量采用**static final修饰**，且用**下划线\_分隔，全大写**
13. 所有的输入参数均采用**final修饰，**防止在方法的内部被修改调用
14. 所有变量命名尽量不要使用XXXList XXXMap, XXXSet等命名方式，一般XXXs即可，只有在同一个业务对象有多个相同参数表示的使用才以list、 array、 set、 map进行区分
15. public在protected之前，protected在private之前，属性在方法之前， 常量在普通属性之前
16. 变量命名严禁使用**无意义的字符** 如 a b c等，只能在循环中使用i,j,k
17. 所有的方法行数不能超过限制长度
18. 方法输入个数不能超过**4个**，超过4个需要使用**变量对象**进行包装
19. 控制层方法用**名词**，业务层方法用**动词**
20. **DAO:**
21. 异常：不捕获异常，只需要往外抛出即可
22. 所有表格、列表查询均需排序
23. **SERVICE：**
24. 获取用户信息：均在service层从ThreadLocal中获取
25. 所有的service方法需要单元测试，包括空值测试，范围外值测试，范围内值测试，异常情况测试，以及极值测试。从类继承即可
26. 所有的service层代码需要输入参数检测，日志输入参数，过程日志，日志界别为**INFO**, 且在每条日志最前面加上功能标识，以短横线分隔
27. 异常：不捕获异常，只需要往外抛出即可
28. 查询方法以query、find、search等开头（查询全部以List结尾 queryPeopleList）
29. 增加、修改、删除以insert、add、update、alter、del、delete等开头
30. **Controller:**
31. 参数需要进行**非空检测**
32. 所有的Controller代码无任何业务，且需要捕获异常Throwable，且输出异常日志
33. 控制层采用**两级路径**，短横线分隔，一个**短横线**表示一个**目录**，所有前后端代码结构目录一致
34. NameSpace与控制层两级路径中的第一级相同
35. result中name与jsp必须相同
36. 验证的方法名XXX-judge-XXX
37. 除字典以外的下拉框、单选框等数据的加载XXX-list-load-data
38. **属性为数据加载XXX-tree-load-data**
39. 已有变量若无法满足需求，若需要定义变量，要讨论后进行定义
40. **日志示例**

**以用户管理为例：**

              1）Controller日志

                LOGGER.error(“usermanagement-action error.”,e);

              2）Service层日志

[LOGGER.info](http://logger.info)(“usermanagement-service input param:[userId“ + userId);

              3）DAO层日志

[LOGGER.info](http://logger.info)(“usermanagement-dao input param:[userId“ + userId);

1. **Maven发布：**

            框架提供的maven包需要发布源代码与javadoc到maven仓库上

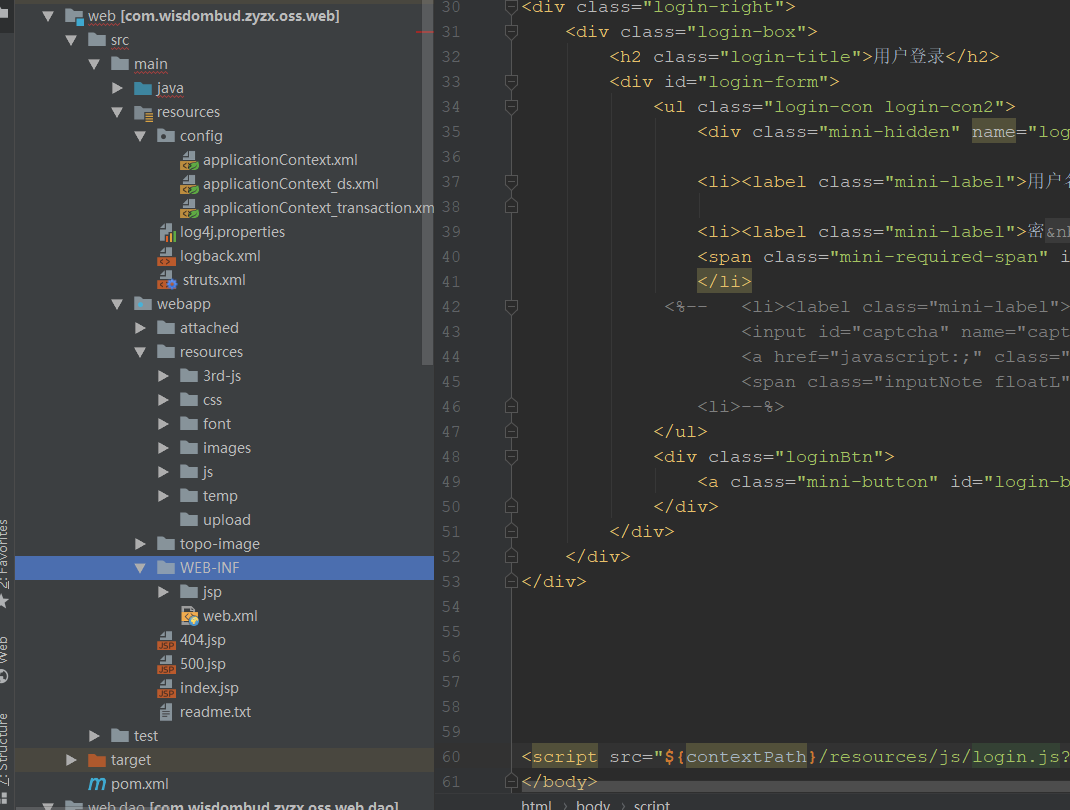
## **后台**

1. **表结构：**
2. 每个的大文本建立单独的表，与基本表建立一对一关系
3. 所有的附件均存储到附件表BASE\_ATTACHMENT
4. 多对多关系以REL开头,SEC表除外
5. 所有的表结构，**初始数据**变动需要**同步到脚本**中，具体由项目组内一个单独的人员负责更新脚本，脚本需要使用脚本测试库进行测试
6. **数据库字段属性最大存储1000字节，超出的一律新建大文本表**
7. 基本表**伪删除**：使用两个字段DICT\_DELETED\_NAME,DICT\_DELETE\_VALUE
8. **格式：**
9. 所有人员的使用相同的格式化代码配置，宽度不超过120
10. 只有一行的代码块不能缺少{}
11. **变量命名:**
12. 后端代码中的枚举使用枚举模板类定义
13. 常量采用static final修饰，且用下划线\_分隔，全大写
14. 所有的输入参数均采用**final修饰**
15. 所有变量命名尽量不要使用XXXList XXXMap, XXXSet等命名方式，一般XXXs即可，只有在同一个业务对象有多个相同参数表示的使用才以list array set map进行区分
16. public在protected之前，protected在private之前，属性在方法之前， 常量在普通属性之前
17. 变量命名**严禁使用无意义的字符** 如 a b c等，只能在循环中使用i,j,k
18. 所有的方法行数不能超过限制长度
19. 方法输入个数**不能超过4个**，超过4个需要使用变量对象进行包装
20. **多线程**
21. 后端代码多个线程修改同一个变量要**锁定同一个监视器**
22. **多线程访问考虑使用volatile, 原子变量,LockFree, ConcurrentHashMap容器**等
23. 尽量使用多线程容器类WorkerManager WorkerManagerFactory
24. **日志:**

        日志要尽量详细，特别是过程日志

## **文件位置**

（以idea为例）



1. src/main/java 存放java代码
2. src/main/java/resources 存放配置文件
3. src/webapp/resources下存放静态资源
   * 1. 3rd-js下存放第三方js
     2. css下存放样式文件
     3. images存放图片
     4. js下存放自定义js脚本
4. src/webapp/WEB-INF/下存放web.xml
5. src/webapp/WEB-INF/jsp下存放自定义jsp文件

公共jsp即首页、错误页面存放在WEB-INF下