OA服务云

开发规范

2018年4月

**更新记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **更新日期** | **说明** | **更新人** |
| v1.0 | 2018-04-09 | 初稿 | 余磊 |
| v1.1 | 2018-05-10 | 增加“2.3url资源组织”小节  6.1命名规范，数据库表名规则与视图名规则修正。 | 余磊 |
|  |  |  |  |

**目录**

[1. 概述 1](#_Toc513712605)

[1.1. 目的 1](#_Toc513712606)

[1.2. 范围 1](#_Toc513712607)

[2. 构建规范 1](#_Toc513712608)

[2.1. 项目目录结构 1](#_Toc513712609)

[2.2. java命名空间 3](#_Toc513712610)

[2.3. url资源组织 4](#_Toc513712611)

[2.4. 配置文件组织 4](#_Toc513712612)

[2.5. 产品内部组织样例 5](#_Toc513712613)

[3. 界面规范 7](#_Toc513712614)

[3.1. 界面布局 7](#_Toc513712615)

[3.2. 表格 7](#_Toc513712616)

[3.3. 数据展现 8](#_Toc513712617)

[3.4. 提示信息 8](#_Toc513712618)

[4. 代码规范 9](#_Toc513712619)

[5. 应用规范 9](#_Toc513712620)

[5.1. 代码警告 9](#_Toc513712621)

[5.2. 日志 9](#_Toc513712622)

[5.3. 金额计算 10](#_Toc513712623)

[5.4. 安全性 10](#_Toc513712624)

[5.4.1. SQL注入攻击 10](#_Toc513712625)

[5.4.2. XSS跨站脚本攻击 10](#_Toc513712626)

[5.4.3. 上传和下载 10](#_Toc513712627)

[6. 数据库规范 11](#_Toc513712628)

[6.1. 命名规范 11](#_Toc513712629)

[6.2. 应用规范 11](#_Toc513712630)

# 概述

## 目的

为了在OA云产品平台开发过程中，提高软件开发质量，降低开发和维护成本，加快交付速度，特制定此规范。

本文适用于产品开发负责人和开发团队成员阅读，并遵照执行。

带（#）号的规范为推荐做法，不强制执行。

## 范围

本规范应用范围，为OA服务云下所有产品/模块。

# 界面规范

软件界面展现，如果没有专门的需求说明，应当遵循以下规范。

此处软件界面展现，应当包括界面显示、导出、打印等所有输出。

## 界面布局

* 界面布局以主流电脑显示分辨率为准，需保证1024\*768（台式机）和1280\*800（笔记本）这两个分辨率下的布局合理性与美观。最低应当支持1024\*768这个分辨率。
* 输入组件大小应当适应正常输入内容的长度，一般情况下不应在输入组件内出现滚动条。
* 多个组件上下排列，组件左右两端应当对齐。
* 文字标签和输入组件左右排列，文字标签应当靠右对齐。
* （#）必填字段的文字标签中，应当使用红色“\*”（半角）表示。

## 表格

* 表格最左侧应当有序号列。如果是分页数据表格，序号应当是整体序号，而不是当前页的序号。
* 表格列标题，应当居中。适当安排列宽，尽量不要让列标题文字换行。
* （#）对表格数据的操作按钮，不应放在表格的某一列内。
* （#）分页数据表格，分页栏应当提供每页行数选择。选项可根据该表格数据量和性能要求来确定。
* 界面显示的列表数据应当默认按照某种业务规则排序（例如编号、时间顺序），不应使用数据库存储顺序。
* （#）分页数据表格，如果允许点击列头排序，则使用全量数据排序，而不是仅对当页数据排序（易造成使用者误解）。如未提供全量数据整体排序功能，则不允许点击列头排序。
* （#）表格应当尽量提供双击操作。可以是双击修改或查看明细。

## 数据展现

数据展现包括在界面上数据输入、选择组件或表格。

* 文本字符串左对齐。
* 数值（包括整数、浮点数、百分数、比率等）靠右对齐。
* 金额应当为金额格式，有千分位并保留2位小数，靠右对齐。
* （#）百分比数值应当直观显示百分号。而不是仅仅在标签或列头上用%说明。如要显示3%，不应仅仅显示3，或直接显示0.03。
* （#）日期、时间应当按统一的格式显示（YYYY-MM-DD或YYYY-MM-DD HH:MI:SS）。在表格列中建议居中。
* （#）枚举字段，如状态、是否、启用等，并且长度较短且字段长度基本一致的字段，在表格列中可设置为居中。

## 提示信息

* 对数据的更新操作，如修改、删除之前，必须要有确认步骤，经操作者确认后再执行操作。新增数据操作一般不需要确认。
* 显示给用户的提示信息，不应有类似指责的强烈语气，也不要用感叹号。
* 业务提示信息，不应当出现英文或其他外语字符。

# 代码规范

各产品/模块的代码规范，由各自开发经理负责制定。

代码规范主要包括java后端代码和web前端代码，涉及到代码的命名规范、代码风格等方面。

代码规范制定时，应当遵循以下原则：

* 有利于提高开发效率，提高协同工作效率
* 有利于提高交付质量
* 有利于提高维护性
* 产品内部，代码风格应尽量保持一致

# 应用规范

## 代码警告

* 最终交付的Java代码中，应当避免出现警告。对于各类警告，可以通过合适的注解来避免。

## 日志

* 最终交付的Java代码中，不应使用System.out输出，因为强制输出会降低性能。应通过日志组件输出。推荐使用logback，也可使用log4j。

## 金额计算

* 涉及到高精度的浮点计算，java代码推荐使用BigDecimal类型变量进行计算。
* 高精度浮点数值，MySQL数据库推荐使用decimal类型字段存储，Oracle数据库推荐使用Number类型字段存储。
* 尽量避免在前台界面，使用js进行高精度金额计算。

## 安全性

开发过程需要考虑安全性

### SQL注入攻击

* 不允许前台拼写SQL语句，所有SQL语句必须在Java后台拼写。
* 后台执行的SQL不允许手工拼接传入的参数，必须使用参数化SQL。

### XSS跨站脚本攻击

* 为了避免XSS攻击，前台后台交互，不允许传输以下特定符号：<、>、[、]。
* 如果前台界面录入了这些符号，应当提示使用者不允许输入。
* 服务端接受到此类数据，应直接过滤掉。

### 上传和下载

* 提供的下载资源，不应直接通过资源的url链接下载。而应该通过一个action，写入Response提供下载。在下载action，应当限制下载内容范围。
* 上传文件（数据模板、附件）等，应当限制文件类型和大小。

# 数据库规范

OA服务云，使用MySQL5.6或更高版本数据库。

以下所有规范都是基于此版本。

## 命名规范

为了保证各产品/模块的数据库对象不冲突，需定义一套命名规范。

首先需要确定各产品的英文缩写。以下示例，假设模块缩写为BS。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **对象类型** | **对象类型英文** | **命名规则** | **示例** |
| 1 | 表 | Table | T + 模块 + \_ + 表名 | TBS\_Station |
| 2 | 主键 | Primary Key | 无需名称 |  |
| 3 | 外键 | Foreign Key | 模块 + \_FK\_ + 外键名 | BS\_FK\_Station\_Type |
| 4 | 索引 | Index | 模块 + \_IX\_ + 索引名 | BS\_IX\_Station\_Code |
| 5 | 视图 | View | V + 模块 + \_ + 视图名 | VBS\_Unit |
| 6 | 函数 | Function | 模块 + \_FUN\_ + 函数名 | BS\_FUN\_GetItemID |
| 7 | 存储过程 | Stored Procedure | 模块 + \_SP\_ + 过程名 | BS\_SP\_CreateNewName |
|  |  |  |  |  |

* 如果视图是为了向下兼容、系统集成等原因建立，可根据实际需要命名。

## 应用规范

* 数据库表涉及，首先应当满足数据库范式，再针对业务进行优化。
* 不应使用有业务含义的字段作为主键。
* （#）不建议使用触发器
* （#）不建议使用自增长序列字段