### 概述

#### 目的

提升代码质量，提高阅读性、复用性，加强团队合作能力，提升项目效率。

此文档不定期更新，会从更多的优秀开源项目中，汲取更优的编码方案，逐步提升个人及团队能力。

#### 适用范围

项目参与者需遵守此约定。

#### 文件及资源命名

资源文件命名全部使用小写字母加驼峰命名方式。

js文件压缩后的min、或cdn混淆后的前缀等，使用点分割（.）

### eslint部分

项目中使用了eslint进行约束，所有项目参与者需遵守其规则，不得私自修改相关配置。

特殊情况如：对外部js引用后，生成了自定义对象，可直接在上下文中使用实例，如果eslint报出“未定义”错误，推荐两种方式解决：

1. 修改引入的js文件，使其支持export方式；
2. 如是在线的js，比如埋点、地图等，可加入“/eslint-disable-line”进行例外处理；

##### 命名规则

Camel Case 和 Pascal Case 均表示“驼峰式大小写”；

区别为Camel Case命名法是由小写字母开始，后续每个单词首字母大写；

Pascal Case以大写字母开头，后续每个单词首字母大写；

1. 变量：Camel Case，命名请不要使用动词
2. 函数：Camel Case，命名请以动词开头

|  |  |
| --- | --- |
| **动词** | **含义** |
| can、has、is | 函数返回一个布尔值 |
| get | 函数返回一个非布尔值 |
| set | 函数保存一个值 |

1. 常量：请以全部大写的形式
2. 构造函数：Pascal Case，名词形式通过此约定可帮助我们快速定位，如果遇到以此形式命名的函数，前面一定要有new运算符。
3. 字符串：使用单引号（单双引号并无不同，此约定是为保持整体风格）

##### 关注点分离

这里的关注点主要指的是：信息（HTML 结构）、外观（CSS）和行为（JavaScript），严格地保证三者分离。

1. 不使用行内样式（<style>.no-good {}</style>）
2. 尽量不在元素上使用 style 属性（<hr style="border-top: 5px solid black">）（动画的transform除外）
3. 不使用行内脚本（<script>alert('no good')</script>）
4. 不使用表象元素（<b>, <u>, <center>, <font>, <b>）
5. 不使用表象 class 名（red, left, center）
6. 所有非公共class必须限制在当前.page中，如：当前有<div class=“page page-demo”>，所有当前页的class一定放在.page-demo{}中，防止污染全局；、
7. 数字：不要省略整数部分或小数部分
8. null：请区分与undefined区别，不要使用null检测未初始化的变量以及传入的参数、
9. undefined：尽量避免使用undefined如无特殊情况，不推荐项目中使用undefined校验。如需判断变量，请使用null方式占位。

##### 注释

1. 模块注释：模块文件首行必须有说明注释， 目的是说明当前模块的描述和依赖关系等。
2. 单行注释：请在复杂逻辑或关键逻辑节点进行标注，注意缩进与代码保持一致
3. 多行注释：方法前使用“ /\*\* \*/“说明函数，便于折叠后能看到每个方法的简要说明。

##### CSS部分

1. 命名:请避免使用难以理解的class名称， class命名请以中划线 .title-color{ }， id命名请以下划线或驼峰方式 ps: #menu\_list || #menuList
2. 不要使用ID，不仅是因为复用性，使用ID后会影响选择器的权重，使得代码变得很奇怪
3. 请尽可能精确的使用选择器 可以这样 .content > .title { }

尽量不要这样： .content .title { }

1. 如果值为0，请不要加单位
2. 尽量使用3个字符的十六进制表示法 可以这样： div { color: #fff; } 别这样：div { color: #ffffff; }
3. 声明顺序：这是一个选择器内书写CSS属性顺序的大致轮廓。这是为了保证更好的阅读体验。作为最佳实践，我们应该遵循以下顺序: **结构属性：**  a) display b) position, left, top, right etc. c) overflow, float, clear etc. d) margin, padding **表现性属性**  a) background, border b) font, text
4. sass中请使用选择器嵌套，但请尽量避免空选择器的嵌套,并注意嵌套中空行的处理
5. DOM结构不要超过8层，css结构尽量不要超过3层

##### 路由名称

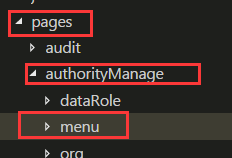
为了在开发中更好的定位问题及快速找到页面，路由名称以pages目录以下的路径名称为路由名称

ps：

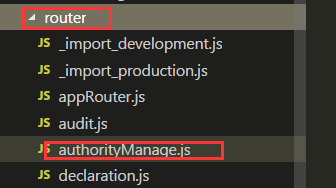
比如权限管理目录下的菜单管理页面



对应pages下的路由名称及目录结构为：



对应的路由名称为：





##### 相关约定

新建目录，路由文件，api请求文件，为避免文件过大及开发时冲突，定以下约定

比如菜单为：

权限管理/菜单管理

对应： pages/ authorityManage/menu

对应： router/ authorityManage.js

对应： js/api/ authorityManage

##### 目录结构及功能

|- project/

|-build/ webpack开发配制，一般不需要动

|-config/ webpack开发配制，一般不需要动

|-dist/ 构建成功后的目录文件

|-src/

|-assets/ 存放静态文件

|-components/ 公共组件

|-filters/ 过滤器

|-js/ api请求和工具类

|-pages/ 页面

|-router/ 路由相关配制

|-scss/ scss相关文件，公共样式，变量，重置文件

|-store/ 公用数据管理一般不需要动

|-vendor/ 封装类组件

|-App.vue vue入口文件

|-busEvent.js vue单例管理

|-config.js 开发配制文件

|-main.js 项目js入口文件

|-mixin.js vue全局公用文件

|-static/ 静态文件，打包时会自动复制到dist

|-.babelrc 配制babel需要的文件

|-.editorconfig 开发格式文件

|-.eslintignore eslint配制文件，用于排除，最好不要去改动

|-.eslintrc.js eslint配制文件，最好不要去改动

|-index.html 入口文件，webpackHtmlPluging会做处理不需要动

|-npm-shrinkwrap.json 版本锁定文件，如无下载不需要更新提交

|-yarn.lock 版本锁定文件，如无下载不需要更新提交

|-package.json 工程依赖相关包