技术中心

前端开发指南（vue版）

v1.0.1

湖南视拓信息技术股份有限公司

文档信息

|  |  |
| --- | --- |
| 文档标识 | 待定 |
| 文档名称 | 前端开发指引（vue版） |
| 文档状态 | 正式发布 |
| 适用项目 | 所有VUE前端项目 |
| 文档版本 | 1.0.1 |
| 部 门 | 技术中心 |
| 作 者 | 欧阳龙蛟 |
| 参 与 者 |  |
| 完成日期 |  |
| 备 注 | 前端开发人员请务必仔细阅读该文档 |

历史版本

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 完成日期 | 备注 |
| 1.0.0 | 欧阳龙蛟 |  | 2017-09-08 | 新建 |

[1前言 4](#_Toc23106)

[2技术栈 4](#_Toc8370)

[2.1基础 4](#_Toc965)

[2.2 vue生态 4](#_Toc19602)

[2.3包管理 4](#_Toc7306)

[2.4构建工具 5](#_Toc27699)

[2.5异步方案 5](#_Toc17016)

[2.6其它 5](#_Toc9243)

[3项目搭建 5](#_Toc30027)

[3.1安装node.js 5](#_Toc29984)

[3.2安装vue-cil 5](#_Toc23143)

[3.3安装项目 5](#_Toc32622)

[3.3.1执行步骤 5](#_Toc18487)

[3.3.2修改配置 6](#_Toc25328)

[3.3.3一般错误 6](#_Toc7551)

[3.4安装项目依赖包 6](#_Toc20431)

[3.5运行发布 7](#_Toc19537)

[3.6搭建svn上下载的项目 7](#_Toc12329)

[4项目结构 7](#_Toc32122)

[4.1项目目录结构 7](#_Toc23591)

[4.2项目结构说明 8](#_Toc6319)

[4.3业务模块结构 8](#_Toc5147)

[4.4业务模块结构说明 8](#_Toc7868)

[5编码规范 9](#_Toc6231)

[5.1基本规范 9](#_Toc28286)

[5.2 vue组件编码规范 9](#_Toc13519)

[5.2.1 html规范 9](#_Toc4731)

[5.2.2 javascript规范 9](#_Toc1740)

[5.2.3 css规范 9](#_Toc25672)

[5.3 eslint代码检查 9](#_Toc13627)

[5.3.1检查规则 9](#_Toc31779)

[5.3.2配置 10](#_Toc22612)

[5.3.3要求 10](#_Toc22130)

[6通用样式 10](#_Toc29623)

[6.1 common.css 10](#_Toc9721)

[6.2 style.css 11](#_Toc15920)

[6.3其它 11](#_Toc16649)

[7 JavaScript工具类 11](#_Toc3253)

# 1前言

vue项目技术选型、开发环境搭建、项目安装、编码规范、通用样式设定及说明、JS工具类说明、交互要求、通用组件等。

前端开发人员进入开发前，请务必仔细阅读该文档，开发过程中请务必遵循该文档相关条列。

# 2技术栈

## 2.1基础

hml/html5、css/css3、es5/es6

参考资料：

《ECMAScript 6 入门》 <http://es6.ruanyifeng.com/#README>

## 2.2 vue生态

vue2.x <https://cn.vuejs.org/>

vue

vue-router2 <https://router.vuejs.org/zh-cn/>

路由

vuex <https://vuex.vuejs.org/zh-cn/>

状态管理

vue-cli <https://www.npmjs.com/package/vue-cli>

脚手架

element-ui 2 http://element.eleme.io/2.0/#/zh-CN/component/installation

饿了么UI

## 2.3包管理

node.js <https://nodejs.org/en/>

node.js 的包管理器 npm，是全球最大的开源库生态系统

## 2.4构建工具

webpack <http://webpack.github.io/docs/>

前端模块加载器兼打包工具

## 2.5异步方案

axios <https://www.npmjs.com/package/axios>

axios（~~阿克西奥斯~~）基于 Promise 的 HTTP 请求客户端。

## 2.6其它

~~jquery2~~

略

vue-quill-editor https://www.npmjs.com/package/vue-quill-editor

文本编辑器

echarts http://echarts.baidu.com/index.html

可视化数据组件

swiper http://www.swiper.com.cn/

轮播插件

# 3项目搭建

## 3.1安装node.js

下载（https://nodejs.org/en/）安装，略。

（可选）设置npm 淘宝代理镜像（npm -> cnpm）：

npm install cnpm -g --registry=https://registry.npm.taobao.org

## 3.2安装vue-cil

npm install -g vue-cli

## 3.3安装项目

### 3.3.1执行步骤

1、进入项目所在目录

cd {path}

2、新建vue项目{projectName}

vue init webpack {projectName}

3、进入项目

cd {projectName}

4、安装

npm install

说明：

1）{path}为项目的物理地址，如：E:\vue\_webpack\oa.icore.web；

2）{projectName}为项目名称。

### 3.3.2修改配置

1、/build/utils.js

publicPath: '../../' -->打包后背景图片路径问题

（由负责人或架构人员修改）

2、/config/index.js

dev --> 端口

autoOpenBrowser --> 是否自动打开浏览器

（自定义设置）

### 3.3.3一般错误

1、npm install 时报 phantomjs-prebuilt@2.1.15 安装失败

解决方案：

单独安装 npm install phantomjs-prebuilt@2.1.15 --ignore-scripts

## 3.4安装项目依赖包

安装 VueRouter2

npm install vue-router -S

安装 axios

npm install axios -S

安装 vuex

npm install vuex -S

安装 element-ui

npm install element-ui -S

npm install element-ui@next -S

（npm install iview -S）

安装文本编辑器

npm install vue-quill-editor -S

安装echarts

npm install echarts -S

安装babel-polyfill

npm install babel-polyfill -S

（以上安装无顺序要求）

## 3.5运行发布

1、运行

npm run dev // 默认打开[http://localhost:8080/#/](http://localhost:8088/#/)

2、打包发布

npm run build

## 3.6搭建svn上下载的项目

1. 下载SVN上的项目；
2. 在项目的上一级目录依次执行步骤3.1-3.4（其中{projectName}为项目名称，覆盖即可）；
3. 用SVN上的文件替换本地文件（相关文件、目录还原）。

# 4项目结构

## 4.1项目目录结构

/build // webpack配置文件

/config // congfig信息

/dist // build上线项目文件

/node\_modules // 依赖模块

/src // 源码目录

/assets // 资源

/js // 项目JS

/css // 项目样式

/img // 项目图片

/components // 组件

/common // 公共组件

/... // 业务模块组件

... // main.vue/login.vue等组件

/config // 配置文件

/router // 路由

/store // vuex状态管理

/utils // 工具类

App.vue // 页面入口组件

main.js // 应用入口文件

/static // 静态资源、公用资源、第三方库等

/js // 跨项目工具类或方法，common.js

/css // 跨项目样式文件，common.css

/img // 跨项目图片

/lib // 第三方库，如jquery、swiper等

/test // 测试

.babelrc // ES6语法编译配置

.editorconfig // 代码编写规格

.eslintrc.js // 代码检查配置文件

.gitignore // 忽略的文件

favicon.ico // 页面左上角小图标

index.html // 入口html文件

package.json // 项目及工具的依赖配置文件

README.md // 说明

## 4.2项目结构说明

1、此项目结构由负责人/架构人员配置好，一般开发人员只需操作src下相关文件；

2、/src/utils下为本项目的工具类或方法，跨项目的通用类或方法归档至/static/js/目录下；

3、/src/assets下的js/css文件在main.js里以import方式引入；

4、/static下的js/css文件在index.html中以普通方式引入；

5、待补充（容我三思）。

## 4.3业务模块结构

例如论坛模块（forum）：

/src

/components

/forum // 论坛模块

/qustion // 问答子模块

/vote // 投票子模块

/topic // 话题子模块

forum.vue // 论坛模块主组件

## 4.4业务模块结构说明

1、模块主组件中一般包含：

·该模块的子模块的路由入口；

·该模块公用样式。

1. 子模块一般情况下不应该包含子路由，尽量在一个spa中完成。

# 5编码规范

## 5.1基本规范

见《WEB前端开发规范-v2.0.docx》

## 5.2 vue组件编码规范

### 5.2.1 html规范

1、html代码包含在<template>...</template>之中；

2、顶层标签有且只有一个，必须是块级元素，div/ul等均可。

### 5.2.2 javascript规范

1. export default属性顺序为：name、props、components、data、created、mounted、methods、completed、watch、filters和其它；
2. 编码中应注意js对象改变的频次，避免对象重复响应渲染影响性能；
3. 强制要求使用eslint代码检查，详细见5.3 eslint代码检查。

### 5.2.3 css规范

1、属于该模块的通用样式写入该模块主组件（见4.3-4.4）<style>...</style>之中；

2、该组件自有样式写入<style scoped>...</style>之中；

3、每个模块的样式都留有css class钩子，一般由：公司项目名称简写 + ‘-’ + 模块名称简写 + ‘-’ + 其它构成，如：oa-forum-topic-xxxx或oa-forum-ques-xxx；

·公司项目名称简写：一般为两个小写字母，由负责人或架构人员指定好

·模块名称简写：一般为2-5个小写字母，保证其在项目中的唯一性

·其它：一般为1-5个小写字母，保证其在模块中的唯一性

1. （重要）vue组件中的class name不可层叠/src/assets/css或/static/css下的class name，如有需要，在组件中追加class；
2. 无特殊情况不需要重新定义字体颜色，字体颜色从6.1common.css中获取；
3. 待补充（容我三思）。

## 5.3 eslint代码检查

### 5.3.1检查规则

默认eslint配置，修改点如下：

1、语句结束必须以分号（;）结束；

2、允许连续声明，如：let a = 1, b = 2, ...;

3、允许匿名实例化对象，如：new Object()。

如需调整联系负责人或架构人员。

### 5.3.2配置

1、已在.eslintrc.js中配置好，一般情况下开发人员无需改动，如有特殊情况，需与负责人或架构人员商量；

2、WebStorm设置：File -> Settings -> Lauguages & Framewors -> JavaScript -> Code Quality Tools -> ESLint -> Configuration file: 指向.eslintrc.js文件（需在Enable状态下）。

### 5.3.3要求

1、确保在WebStorm中无error级别（系数为2）错误（红色波浪线）；

2、确保build的时候无error提示；

3、确保兼容IE9+。

# 6交互要求

## 6.1操作交互

### 6.1.1按钮状态

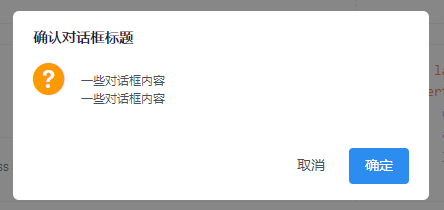
所有登录、保存、提交、删除等有关增删改操作的按钮点击后都需要置为loading状态（失效），如图：

TIM截图20170922160019

操作成功、操作出错后需要将按钮回归到正常状态。

### 6.1.2 删除

所有删除操作必须要有确认框，如图：



## 6.2页面优化

### 6.2.1弹窗展开方向

如日期弹窗、下拉框弹框，默认朝下，如果出现在表单（页面）下面，不够显示的则指定其朝上。

## 6.3其它

# 7通用样式

## 7.1 common.css

归档至 static/css/，内容包括：

1、定义css rest；

2、定义字体颜色



3、定义animation；

4、待补充（容我三思）。

## 7.2 style.css

归档至 src/assets/css，内容为：

1、定义从属项目的css sprites；

2、定义从属项目的字体颜色（在有需要时可覆盖common.css中的字体颜色）；

3、定义从属项目的跨模块通用组件公用css；

4、待补充（容我三思）。

## 7.3其它

1、其它跨项目通用css归档至 static/css/（一般不会有）；

2、其它跨模块通用css归档至src/assets/css，如无特殊情况不需要单独创建，一并写入style.css；

3、项目主题（样式）切换的层叠样式文件归档至src/assets/css；

4、待补充（容我三思）。

# 8 JavaScript工具类

## 8.1 utils

*持续建设中...*

# 9通用组件

局部组件，使用的时候需要import & components

## 9.1上传组件

### 9.1.1基本信息

### 9.1.2使用方法

### 9.1.3 props

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 说明 | 类型 | 默认值 |
|  |  |  |  |

### 9.1.4 emit

|  |  |  |
| --- | --- | --- |
| 事件名 | 说明 | 返回值 |
|  |  |  |

### 9.1.5注意事项

## 9.2文件列表组件

## 9.3人员选择组件

## 9.4日历表组件

## 9.5工作流组件

## 9.6打印组件

### 9.6.1表格打印

### 9.6.2表单打印

*持续建设中...*