web开发规范

说明：该规范适合vue-cli搭建的前端项目，其他项目可大致参考

# **⼀、编码规范**

## **1、源⽂件**

### 1.1⽂件名

vue文件以文件名加上.vue后缀组成

js文件以文件名加上.js后缀组成

json文件以文件名加上.json后缀组成

css文件以文件名加上.css后缀组成

scss文件以文件名加上.scss后缀组成

less文件以文件名加上.less后缀组成

### 1.2⽂件编码

源码⽂件使⽤UTF-8编码。

### 1.3⽂件结构

vue文件结构从上到下顺序依次为：template标签、script标签、style标签

## 2、格式

### 2.1基本规范

a. 缩进使⽤2个空格，禁⽌使⽤Tab（因为tab缩进在不同的操作系统上或在不同的编辑器下显示长短不一定）, ⼀⾏只有⼀个语句；

b. ⼀⾏只声明⼀个变量；

c. 单⾏⻓度不超过120，import语句不受⻓度限制；

d. 换⾏符使⽤格式：\r\n ，对应IDEA、VScode右下角显示为CRLF；

e. 常量使用const声明，全局变量使用var声明，局部变量使用let声明；

### 2.2换⾏

当⼀⾏语句⻓度超过120，使⽤以下规则断开：

a. 逗号后⾯断开

b. 操作符前⾯断开

c. 换⾏缩进使⽤2个空格

### 2.3语句块

语句块指类、⽅法以及if/else/for/do/while/switch/try/catch/finally语句之后的代码块，语句块总是使

⽤”{ }”符号，使⽤规则：

a. 左⼤括号位于起始⾏结尾

b. 右⼤括号换⾏并与起始⾏对⻬

c. 被括语句使⽤⼀个缩进单位

d. 语句块只有单⾏语句也必须使⽤”{ }”

### 2.4空格

a. 所有保留关键字前后使⽤⼀个空格

b. 语句块的”{”前⾯使⽤⼀个空格

c. ⼀元操作符不加空格

d. ⼆元或三元操作符使⽤⼀个空格，除了”.”

e. 逗号、冒号、分号和右括号后⾯使⽤⼀个空格

### 2.5空⾏

空⾏将不同的代码段分隔开，⽤以提⾼可读性：

a. 1.3的各组成部分之间都是⽤⼀个空⾏

b. ⽂件结尾以⼀个空⾏结束

c. vue、js文件的第⼀个成员变量或常量前⾯和最后⼀个成员变量或常量后⾯使⽤⼀个空⾏

d. ⽅法之间使⽤⼀个空⾏

e. 语句块前后各使⽤⼀个空⾏

f. /\*……\*/和//注释之前使⽤⼀个空⾏

g.成员变量之间使⽤⼀个空⾏做逻辑分组（可选）

h.style标签内不同样式名之间使用一个空行

### 2.6 代码格式检查与格式化：vscode+eslint+prettier

在vscode中安装eslint和Prettier-Code formatter插件，使用eslint规范代码书写规范，prettier配置格式化的规则。项目安装依赖库：

eslint

@vue/cli-plugin-eslint

babel-eslint

prettier

eslint-config-prettier

eslint-plugin-prettier

eslint-plugin-vue

eslint检查规则可参考配置文件.eslintrc.js

module.exports = {

  root: true,

  parserOptions: {

    parser: 'babel-eslint',

    sourceType: 'module'

  },

  env: {

    browser: true,

    node: true,

    es6: true

  },

  extends: ['plugin:vue/recommended', 'eslint:recommended', 'prettier'],

  // extends: ['prettier'],

  plugins: ['prettier'],

  // add your custom rules here

  //it is base on https://github.com/vuejs/eslint-config-vue

  rules: {

    'prettier/prettier': 'off',

    'vue/max-attributes-per-line': [

      2,

      {

        singleline: 10,

        multiline: {

          max: 10,

          allowFirstLine: true

        }

      }

    ],

    //"vue/name-property-casing": ["error", "PascalCase"],

    'accessor-pairs': 2,

    'arrow-spacing': [

      2,

      {

        before: true,

        after: true

      }

    ],

    'block-spacing': [2, 'always'],

    'brace-style': [

      2,

      '1tbs',

      {

        allowSingleLine: true

      }

    ],

    camelcase: [

      0,

      {

        properties: 'always'

      }

    ],

    'comma-dangle': [2, 'never'],

    'comma-spacing': [

      2,

      {

        before: false,

        after: true

      }

    ],

    'comma-style': [2, 'last'],

    'constructor-super': 2,

    curly: [2, 'multi-line'],

    'dot-location': [2, 'property'],

    'eol-last': 2,

    eqeqeq: [2, 'allow-null'],

    'generator-star-spacing': [

      2,

      {

        before: true,

        after: true

      }

    ],

    'handle-callback-err': [2, '^(err|error)$'],

    // 'indent': [2, 2, {

    //   'SwitchCase': 1

    // }],

    'jsx-quotes': [2, 'prefer-single'],

    'key-spacing': [

      2,

      {

        beforeColon: false,

        afterColon: true

      }

    ],

    'keyword-spacing': [

      2,

      {

        before: true,

        after: true

      }

    ],

    'new-cap': [

      2,

      {

        newIsCap: true,

        capIsNew: false

      }

    ],

    'new-parens': 2,

    'no-array-constructor': 2,

    'no-caller': 2,

    'no-console': 'off',

    'no-class-assign': 2,

    'no-cond-assign': 2,

    'no-const-assign': 2,

    'no-control-regex': 0,

    'no-delete-var': 2,

    'no-dupe-args': 2,

    'no-dupe-class-members': 2,

    'no-dupe-keys': 2,

    'no-duplicate-case': 2,

    'no-empty-character-class': 2,

    'no-empty-pattern': 2,

    'no-eval': 2,

    'no-ex-assign': 2,

    'no-extend-native': 2,

    'no-extra-bind': 2,

    'no-extra-boolean-cast': 2,

    'no-extra-parens': [2, 'functions'],

    'no-fallthrough': 2,

    'no-floating-decimal': 2,

    'no-func-assign': 2,

    'no-implied-eval': 2,

    'no-inner-declarations': [2, 'functions'],

    'no-invalid-regexp': 2,

    'no-irregular-whitespace': 2,

    'no-iterator': 2,

    'no-label-var': 2,

    'no-labels': [

      2,

      {

        allowLoop: false,

        allowSwitch: false

      }

    ],

    'no-lone-blocks': 2,

    'no-mixed-spaces-and-tabs': 2,

    'no-multi-spaces': 2,

    'no-multi-str': 2,

    'no-multiple-empty-lines': [

      2,

      {

        max: 1

      }

    ],

    'no-native-reassign': 2,

    'no-negated-in-lhs': 2,

    'no-new-object': 2,

    'no-new-require': 2,

    'no-new-symbol': 2,

    'no-new-wrappers': 2,

    'no-obj-calls': 2,

    'no-octal': 2,

    'no-octal-escape': 2,

    'no-path-concat': 2,

    'no-proto': 2,

    'no-redeclare': 2,

    'no-regex-spaces': 2,

    'no-return-assign': [2, 'except-parens'],

    'no-self-assign': 2,

    'no-self-compare': 2,

    'no-sequences': 2,

    'no-shadow-restricted-names': 2,

    'no-spaced-func': 2,

    'no-sparse-arrays': 2,

    'no-this-before-super': 2,

    'no-throw-literal': 2,

    'no-trailing-spaces': 2,

    'no-undef': 2,

    'no-undef-init': 2,

    'no-unexpected-multiline': 2,

    'no-unmodified-loop-condition': 2,

    'no-unneeded-ternary': [

      2,

      {

        defaultAssignment: false

      }

    ],

    'no-unreachable': 'off',

    'no-unsafe-finally': 2,

    'no-unused-vars': [

      2,

      {

        vars: 'all',

        args: 'none'

      }

    ],

    'no-useless-call': 2,

    'no-useless-computed-key': 2,

    'no-useless-constructor': 2,

    'no-useless-escape': 0,

    'no-whitespace-before-property': 2,

    'no-with': 2,

    'one-var': [

      2,

      {

        initialized: 'never'

      }

    ],

    'operator-linebreak': [

      2,

      'after',

      {

        overrides: {

          '?': 'before',

          ':': 'before'

        }

      }

    ],

    'padded-blocks': [2, 'never'],

    quotes: [

      2,

      'single',

      {

        avoidEscape: true,

        allowTemplateLiterals: true

      }

    ],

    semi: [2, 'never'],

    'semi-spacing': [

      2,

      {

        before: false,

        after: true

      }

    ],

    'space-before-blocks': [2, 'always'],

    'space-before-function-paren': [2, 'never'],

    'space-in-parens': [2, 'never'],

    'space-infix-ops': 2,

    'space-unary-ops': [

      2,

      {

        words: true,

        nonwords: false

      }

    ],

    'spaced-comment': [

      2,

      'always',

      {

        markers: ['global', 'globals', 'eslint', 'eslint-disable', '\*package', '!', ',']

      }

    ],

    'template-curly-spacing': [2, 'never'],

    'use-isnan': 2,

    'valid-typeof': 2,

    'wrap-iife': [2, 'any'],

    'yield-star-spacing': [2, 'both'],

    yoda: [2, 'never'],

    'prefer-const': 2,

    'no-debugger': process.env.NODE\_ENV === 'production' ? 2 : 0,

    'object-curly-spacing': [

      2,

      'always',

      {

        objectsInObjects: true

      }

    ],

    'array-bracket-spacing': [2, 'never'],

    'vue/singleline-html-element-content-newline': 'off',

    'vue/html-self-closing': 'off' // 去掉标签需要自闭和的警告

  }

}

prettier配置规则参考.prettierrc.js

module.exports = {

  printWidth: 120,

  tabWidth: 2, // tab缩进大小,默认为2

  useTabs: false, // 使用tab缩进，默认false

  semi: false, // 语句末尾加分号, 默认true

  singleQuote: true, // 使用单引号, 默认false(在jsx中配置无效, 默认都是双引号)

  quoteProps: 'as-needed',

  jsxSingleQuote: true, // 在JSX中使用单引号, 默认false

  trailingComma: 'none', // 行尾逗号,默认es5(2.0.0),可选 none|es5|all

  bracketSpacing: true, // 对象中的空格 默认true

  jsxBracketSameLine: false, // JSX标签闭合位置 默认false

  arrowParens: 'avoid', // 箭头函数参数括号 默认always 可选 avoid| always

  vueIndentScriptAndStyle: false, // 是否缩进Vue文件中的<script>和<style>标记内的代码，默认false

  endOfLine: 'crlf' // 默认lf，可选 lf|crlf|cr|auto

}

### 2.7 eslint例外

特殊情况的js代码，无法按eslint规范执行的，比如：引入微信js sdk文件， 使用wx.config({})等api时， wx对象会报错，可使用 <!-- eslint-disable-next-line -->

标记下一行不受检查；/\* eslint-disable \*/标记整个文件不受检查

## **3、命名**

### 3.1 vue文件

使用首字母大写的驼峰式写法，如：ToDoList.vue

### 3.2自定义组件内的props内的参数

⼩写开头的驼峰式命名：aPropName

### 3.3 js文件

使⽤小写加连字符组成，如：common-util.js

### 3.4 ⽅法

使⽤⼩写开头的驼峰式命名：aFunction

### 3.5变量

使⽤⼩写开头的驼峰式命名：aVariable

### 3.6常量

全部使⽤⼤写字符，单词之间⽤下划线隔开：CONSTANCE\_CASE

### 3.7其它

a. js文件或vue文件名使⽤名词或名词短语，⽅法名使⽤动词开头

b. 禁⽌使⽤拼⾳或中⽂

c. 测试类名使⽤Test结尾，测试⽅法名使⽤test开头

## **4、注释**

### 4.1单行注释

使用// 后带一个空格，如：// 这是单行注释

### 4.2多行注释

使用如下方式：

/\*\*

 \* 这是多行注释第一行

 \* 这是多行注释第二行

 \*/

如是方法注释，需加上参数和返回结果的说明，如：

/\*\*

 \* 这是多行注释第一行

 \* 这是多行注释第二行

 \* paramA:参数a含义（若有）

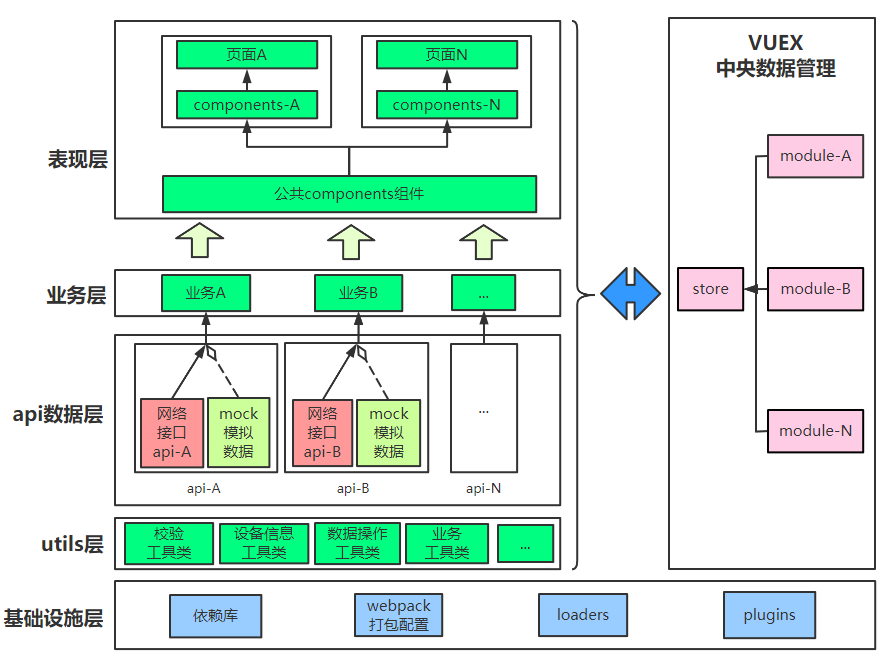
 \* paramB:参数b含义（若有）

 \* return:返回结果含义（若有）

 \*/

# **二、工程规范**

## **1、应⽤分层**



### 1.1 表现层

表现层即UI层为应用所有页面及自定义组件层，包含组件布局和样式。分为公共组件及各页面，各页面内也建议封装功能单一或逻辑复杂的组件来解耦较为复杂的逻辑处理和交互。

### 1.2业务层

由于vue文件的特殊性，业务层位于每个vue功能页面的script标签内逻辑代码，包含交互逻辑、调用业务接口等。

### 1.3 api数据层

数据层为应用所有的接口数据api访问集合、以功能模块对数据层进行分类，

每个功能模块的数据层可由2个数据源提供数据：

网络数据源：真实的网络请求数据

mock数据源: 开启mock服务拦截网络接口模拟接口的返回数据，响应页面变化，应用于网络接口未开发完、或者模拟不同的接口返回结果测试页面应对各种情况的响应状态。

### 1.4 utils层

Utils层为应用所有的工具类集合，按类别进行划分，如：校验类的工具类、时间处理相关的工具类、设备信息相关的工具类、业务相关的工具类等等，实现将工具层模块化。

### 1.5 基础设施层

基础设施层使用webpack构建工程项目，集中式的管理依赖库、使用各类loaders处理不同类型的文件,使用合适的plugins处理打包过程的优化以及文件处理等。

### 1.6 中央数据管理

使用vuex来做全局数据管理以及数据的响应变化，定义不同module模块进行数据的分类管理

## **2、代码管理**

### 2.1版本管理

版本仓库统⼀使⽤Git。

本地全局性设置中⽂名和公司邮箱：

git confifig --global user.name “<中⽂名>”

git confifig --global user.email “<公司邮箱>”

提交⽂件范围只应包括：

a. 程序代码

b. 配置⽂件

c. 构建脚本

d. README.md

禁⽌提交以下⽂件：

a. node\_modules依赖库文件夹

b. dist产出文件夹

c. .idea 等IDEA文件

### 2.2格式化

为了确保提交到Git仓库的代码符合第⼀章的编码规范，可以使⽤以下方法进⾏辅助：

a. ESlint规范代码格式。

b. 自己编码的部分（他人代码不要轻易改动）使用代码格式化后再提交

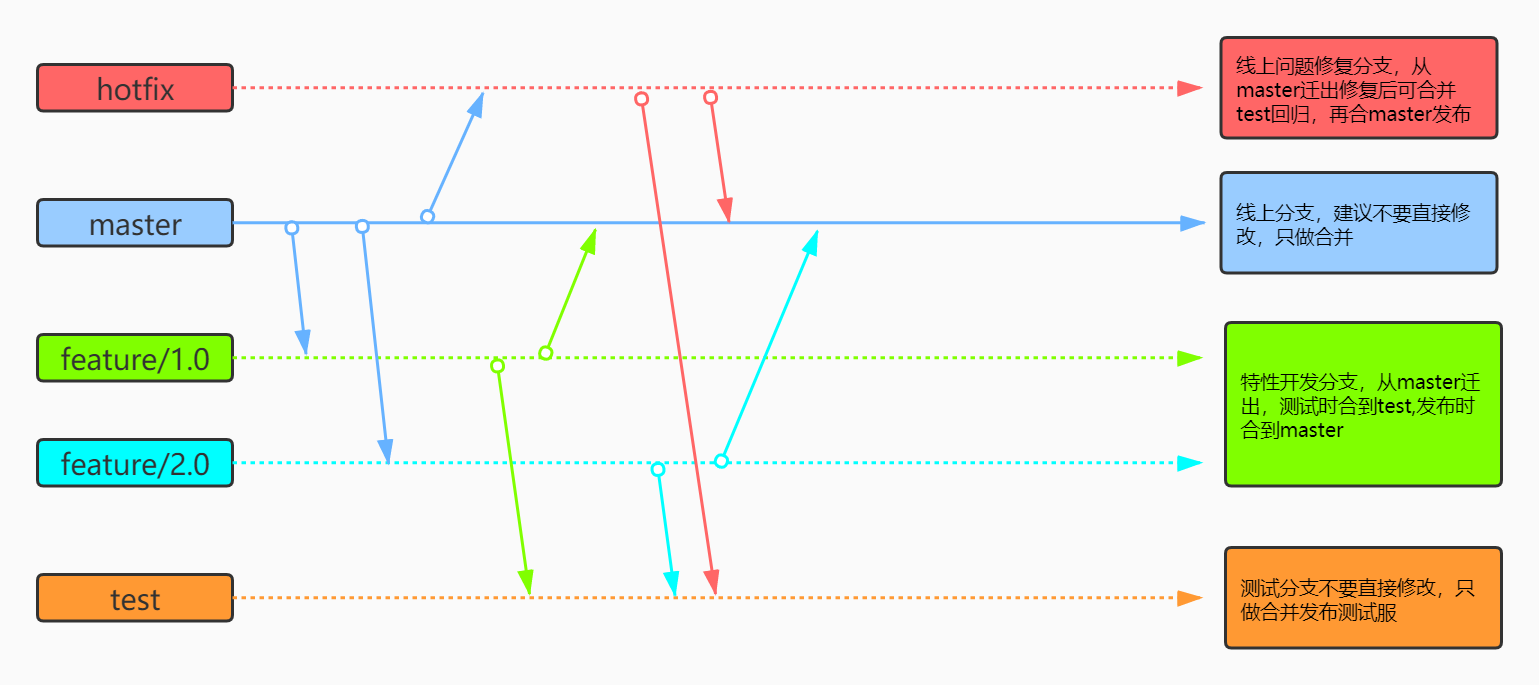
## **3、Git分⽀模型**

使⽤主流分⽀模型Gitflflow，包括以下分⽀类型：

master：主分⽀，远程分⽀，⽤于发布版本，禁⽌直接提交

feature分⽀：特性开发分支，从master迁出，测试时合到test，发布时合到master

test分⽀：测试分支，不要直接修改，只做合并发布测试服

hotfix分⽀：线上问题修复分支，从master迁出修复后可合并test回归，再合master发布

# **三、构建版本**

建议构建脚本命名如下：

## **1、本地调试**

npm run dev

## **2、构建开发版本**

npm run build:dev 对应产出目录outputDir建议设置为dist-dev

## 3、构建测试版本

npm run build:test 对应产出目录outputDir建议设置为dist-test

## 4、构建生产版本

npm run build:prod 对应产出目录outputDir建议设置为dist-prod