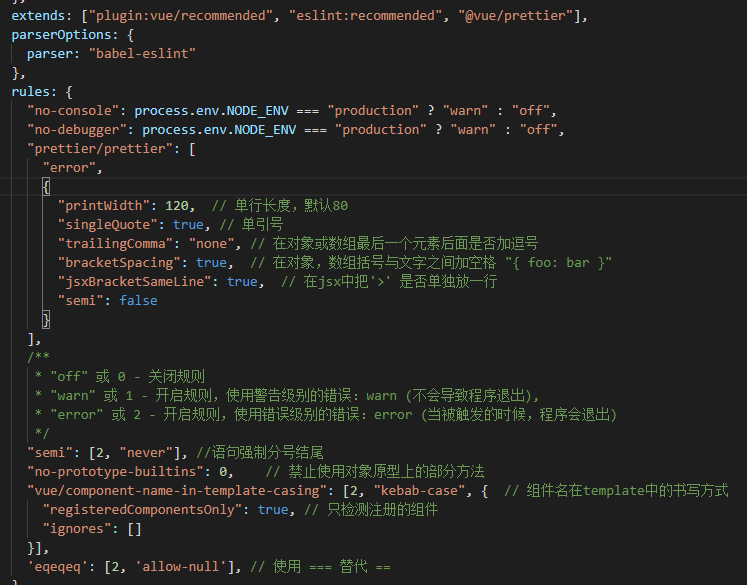
## 规范说明

本规范规定了使用JavaScript、Vue进行开发时的一些命名以及编码规范。使用eslint，推荐如下配置。一些可修复，比较基础的未一一列出，请参考eslint，vue官方文档，并配置好工具的保存自动格式化代码。



## 命名规范

所有的命名应用英文描述其含义，禁止使用拼音，中文简写。长度不宜过长。可适当使用简写。一般情况下，js变量/文件名一般使用camelCase(驼峰命名),html中属性，类名使用kebab-case(短横线分割)

1. 常量使用全大写，单词之间用\_连接
2. 类名、构造函数、公共对象实例等名称PascalCase(首字母大写)
3. boolean类型变量使用"is"。"has","can","should"同理 例：let isShow = true
4. 事件监听函数使用on 例：function onSendMsg() {} （也可使用handle之类，但要让人明白这是一个事件监听的处理函数）
5. 作用域不大临时变量可以简写，比如：str，num，bol，obj，fun，arr。I,j,k（非强制）
6. 组件名采用PascalCase，其name与文件名采用相同的书写格式。尽量名称保持一 致，在HTML中使用组件时，使用kebab-case 方式
7. 组件名应避免与HTML原有标签命名冲突，不确定的可以使用特殊前缀或多个单词
8. 应用特定样式和约定的基础组件 (也就是展示类的、无逻辑的或无状态的组件) 应 该全部以一个特定的前缀开头，比如 Base、App 或 V BaseButton
9. 和父组件紧密耦合的子组件应该以父组件名作为前缀命名 TodoList 、TodoListItem

## 注释说明

良好的注释，能方便我们在代码维护和团队协作中起到很大的作用，在编写代码时，请不要吝啬你的注释

1. 除开一些功能明了，过度统一的文件，所有文件头部必须添加注释，vscode中，可 以使用这个插件进行注释koroFileHeader
2. 变量的注释一般放在其后，当对多行对象进行说明或单行长度过长时，可以选择放 在上一行
3. 方法的注释，需要说明方法用途，入参，出参
4. 在复杂的逻辑、临界、特殊情况的时候，需要添加注释说明

## 常见js书写规范

1. 使用箭头函数代替 let \_this = this
2. console.log(),debugger调试完成后要删除
3. if,else，禁止使用简写，必须跟{}
4. 表达式和语句应清晰、简洁，易于阅读和理解，避免使用晦涩难懂的语句。使用圆括号明确表达式执行优先级
5. 无特殊情况时，使用=== 和 !==进行条件比较
6. 一行代码字符不超过120个字符，一个方法内一般不超过80行，文件中的代码行，不强制要求，一般不超过200行，在复杂的业务页面中，一般不超过1000行，当代码过长时，需要考虑单独分离一些配置，组件，方法
7. 其他一些规则请参考eslint官方<https://eslint.bootcss.com/docs/rules/>

## Vue常见规范

1. data必须使用函数返回一个对象
2. Props中的参数必须说明类型，是否必填，必要时可以使用validator进行验证参数
3. V-for必须绑定key,v-for应禁止与v-if一起使用
4. 组件/实例的选项编写顺序如下：

副作用 (触发组件外的影响)

el

全局感知 (要求组件以外的知识)

name

parent

组件类型 (更改组件的类型)

functional

模板修改器 (改变模板的编译方式)

delimiters

comments

模板依赖 (模板内使用的资源)

components

directives

filters

组合 (向选项里合并 property)

extends

mixins

接口 (组件的接口)

inheritAttrs

model

props/propsData

本地状态 (本地的响应式 property)

data

computed

事件 (通过响应式事件触发的回调)

watch

生命周期钩子 (按照它们被调用的顺序)

beforeCreate

created

beforeMount

mounted

beforeUpdate

updated

activated

deactivated

beforeDestroy

Destroyed

1. 元素 (包括组件) 的 attribute 顺序如下：

定义 (提供组件的选项)

is

列表渲染 (创建多个变化的相同元素)

v-for

条件渲染 (元素是否渲染/显示)

v-if

v-else-if

v-else

v-show

v-cloak

渲染方式 (改变元素的渲染方式)

v-pre

v-once

全局感知 (需要超越组件的知识)

id

唯一的 attribute (需要唯一值的 attribute)

ref

key

双向绑定 (把绑定和事件结合起来)

v-model

其它 attribute (所有普通的绑定或未绑定的 attribute)

事件 (组件事件监听器)

v-on

内容 (覆写元素的内容)

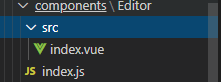
v-html

v-text

1. 其他详细规则参考vue官方<https://cn.vuejs.org/v2/style-guide/>

## 组件、页面开发

1. 组件和页面结构遵循从上往下template，script，style的结构。
2. 为组件根元素设置一个class使用scss加scoped,一般所有样式放在根class下。
3. 一般不使用id，组件中使用id，首选随机id（非强制）
4. 需要外部使用的尽量外部使用ref获取组件。（非强制）
5. 应尽量避免操作dom（非强制）
6. 页面内，或组件内修改公共组件样式时，需要带上自定义的类名来增加命名空 间。全局所有修改时不用添加，一般不建议全局修改
7. 父组件中修改子组件样式时，使用 >>>
8. 页面和组件应遵循以下目录结构，在index中导出组件或页面



import component from './src'

export default component

## 七、git 提交规范

type⽤于说明Commit的类型，包含以下7种类型（非强制，提交文件过多不强制要求，但要写清楚提交的内容）：

1. feat：新功能（feature）
2. fix：修补bug
3. docs：文档（documentation）
4. style： 格式（不影响代码运行的变动）
5. refactor：重构（即不是新增功能，也不是修改bug的代码变动）
6. test：增加测试
7. chore：构建过程或辅助工具的变动

例：git commit -m "feat(views/login): 登录页功能及其接口对接"