目录

[1. eslint规则表 1](#_Toc25530)

[2. eslint忽略检查文件/文件夹 12](#_Toc18082)

[ 忽略文件. 12](#_Toc10291)

[ 忽略文件夹 14](#_Toc1234)

## eslint规则表

* **注意这个规则！！！！：如果违反了规则情况下，这里的数字：0表示不不处理，1表示警告，2表示错误并退出**

**Js行内规则注释（以下是关闭驼峰命名和开启对象属性强制使用引号的规则）**

/\* eslint camelcase: 0, quote-props:[2, "always"] \*/

{"rules" : {

// 定义对象的set存取器属性时，强制定义get

"accessor-pairs": 2,

// 指定数组的元素之间要以空格隔开(,后面)， never参数：[ 之前和 ] 之后不能带空格，always参数：[ 之前和 ] 之后必须带空格

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

// 在块级作用域外访问块内定义的变量是否报错提示

"block-scoped-var": 0,

// if while function 后面的{必须与if在同一行，java风格。

"brace-style": [2, "1tbs", { "allowSingleLine": true }],

// 双峰驼命名格式

"camelcase": 2,

// 数组和对象键值对最后一个逗号， never参数：不能带末尾的逗号, always参数：必须带末尾的逗号，

// always-multiline：多行模式必须带逗号，单行模式不能带逗号

"comma-dangle": [2, "never"],

// 控制逗号前后的空格

"comma-spacing": [2, { "before": false, "after": true }],

// 控制逗号在行尾出现还是在行首出现

// http://eslint.org/docs/rules/comma-style

"comma-style": [2, "last"],

// 圈复杂度

"complexity": [2,9],

// 以方括号取对象属性时，[ 后面和 ] 前面是否需要空格, 可选参数 never, always

"computed-property-spacing": [2,"never"],

// 强制方法必须返回值，TypeScript强类型，不配置

"consistent-return": 0,

// 用于指统一在回调函数中指向this的变量名，箭头函数中的this已经可以指向外层调用者，应该没卵用了

// e.g [0,"that"] 指定只能 var that = this. that不能指向其他任何值，this也不能赋值给that以外的其他值

"consistent-this": 0,

// 强制在子类构造函数中用super()调用父类构造函数，TypeScrip的编译器也会提示

"constructor-super": 0,

// if else while for do后面的代码块是否需要{ }包围，参数：

// multi 只有块中有多行语句时才需要{ }包围

// multi-line 只有块中有多行语句时才需要{ }包围, 但是块中的执行语句只有一行时，

// 块中的语句只能跟和if语句在同一行。if (foo) foo++; else doSomething();

// multi-or-nest 只有块中有多行语句时才需要{ }包围, 如果块中的执行语句只有一行，执行语句可以零另起一行也可以跟在if语句后面

// [2, "multi", "consistent"] 保持前后语句的{ }一致

// default: [2, "all"] 全都需要{ }包围

"curly": [2, "all"],

// switch语句强制default分支，也可添加 // no default 注释取消此次警告

"default-case": 2,

// 强制object.key 中 . 的位置，参数:

// property，'.'号应与属性在同一行

// object, '.' 号应与对象名在同一行

"dot-location": [2, "property"],

// 强制使用.号取属性

// 参数： allowKeywords：true 使用保留字做属性名时，只能使用.方式取属性

// false 使用保留字做属性名时, 只能使用[]方式取属性 e.g [2, {"allowKeywords": false}]

// allowPattern: 当属性名匹配提供的正则表达式时，允许使用[]方式取值,否则只能用.号取值 e.g [2, {"allowPattern": "^[a-z]+(\_[a-z]+)+$"}]

"dot-notation": [2, {"allowKeywords": true}],

// 文件末尾强制换行

"eol-last": 2,

// 使用 === 替代 ==

"eqeqeq": [2, "allow-null"],

// 方法表达式是否需要命名

"func-names": 0,

// 方法定义风格，参数：

// declaration: 强制使用方法声明的方式，function f(){} e.g [2, "declaration"]

// expression：强制使用方法表达式的方式，var f = function() {} e.g [2, "expression"]

// allowArrowFunctions: declaration风格中允许箭头函数。 e.g [2, "declaration", { "allowArrowFunctions": true }]

"func-style": 0,

"no-alert": 0,//禁止使用alert confirm prompt

"no-array-constructor": 2,//禁止使用数组构造器

"no-bitwise": 0,//禁止使用按位运算符

"no-caller": 1,//禁止使用arguments.caller或arguments.callee

"no-catch-shadow": 2,//禁止catch子句参数与外部作用域变量同名

"no-class-assign": 2,//禁止给类赋值

"no-cond-assign": 2,//禁止在条件表达式中使用赋值语句

"no-console": 2,//禁止使用console

"no-const-assign": 2,//禁止修改const声明的变量

"no-constant-condition": 2,//禁止在条件中使用常量表达式 if(true) if(1)

"no-continue": 0,//禁止使用continue

"no-control-regex": 2,//禁止在正则表达式中使用控制字符

"no-debugger": 2,//禁止使用debugger

"no-delete-var": 2,//不能对var声明的变量使用delete操作符

"no-div-regex": 1,//不能使用看起来像除法的正则表达式/=foo/

"no-dupe-keys": 2,//在创建对象字面量时不允许键重复 {a:1,a:1}

"no-dupe-args": 2,//函数参数不能重复

"no-duplicate-case": 2,//switch中的case标签不能重复

"no-else-return": 2,//如果if语句里面有return,后面不能跟else语句

"no-empty": 2,//块语句中的内容不能为空

"no-empty-character-class": 2,//正则表达式中的[]内容不能为空

"no-empty-label": 2,//禁止使用空label

"no-eq-null": 2,//禁止对null使用==或!=运算符

"no-eval": 1,//禁止使用eval

"no-ex-assign": 2,//禁止给catch语句中的异常参数赋值

"no-extend-native": 2,//禁止扩展native对象

"no-extra-bind": 2,//禁止不必要的函数绑定

"no-extra-boolean-cast": 2,//禁止不必要的bool转换

"no-extra-parens": 2,//禁止非必要的括号

"no-extra-semi": 2,//禁止多余的冒号

"no-fallthrough": 1,//禁止switch穿透

"no-floating-decimal": 2,//禁止省略浮点数中的0 .5 3.

"no-func-assign": 2,//禁止重复的函数声明

"no-implicit-coercion": 1,//禁止隐式转换

"no-implied-eval": 2,//禁止使用隐式eval

"no-inline-comments": 0,//禁止行内备注

"no-inner-declarations": [2, "functions"],//禁止在块语句中使用声明（变量或函数）

"no-invalid-regexp": 2,//禁止无效的正则表达式

"no-invalid-this": 2,//禁止无效的this，只能用在构造器，类，对象字面量

"no-irregular-whitespace": 2,//不能有不规则的空格

"no-iterator": 2,//禁止使用\_\_iterator\_\_ 属性

"no-label-var": 2,//label名不能与var声明的变量名相同

"no-labels": 2,//禁止标签声明

"no-lone-blocks": 2,//禁止不必要的嵌套块

"no-lonely-if": 2,//禁止else语句内只有if语句

"no-loop-func": 1,//禁止在循环中使用函数（如果没有引用外部变量不形成闭包就可以）

"no-mixed-requires": [0, false],//声明时不能混用声明类型

"no-mixed-spaces-and-tabs": [2, false],//禁止混用tab和空格

"linebreak-style": [0, "windows"],//换行风格

"no-multi-spaces": 1,//不能用多余的空格

"no-multi-str": 2,//字符串不能用\换行

"no-multiple-empty-lines": [1, {"max": 2}],//空行最多不能超过2行

"no-native-reassign": 2,//不能重写native对象

"no-negated-in-lhs": 2,//in 操作符的左边不能有!

"no-nested-ternary": 0,//禁止使用嵌套的三目运算

"no-new": 1,//禁止在使用new构造一个实例后不赋值

"no-new-func": 1,//禁止使用new Function

"no-new-object": 2,//禁止使用new Object()

"no-new-require": 2,//禁止使用new require

"no-new-wrappers": 2,//禁止使用new创建包装实例，new String new Boolean new Number

"no-obj-calls": 2,//不能调用内置的全局对象，比如Math() JSON()

"no-octal": 2,//禁止使用八进制数字

"no-octal-escape": 2,//禁止使用八进制转义序列

"no-param-reassign": 2,//禁止给参数重新赋值

"no-path-concat": 0,//node中不能使用\_\_dirname或\_\_filename做路径拼接

"no-plusplus": 0,//禁止使用++，--

"no-process-env": 0,//禁止使用process.env

"no-process-exit": 0,//禁止使用process.exit()

"no-proto": 2,//禁止使用\_\_proto\_\_属性

"no-redeclare": 2,//禁止重复声明变量

"no-regex-spaces": 2,//禁止在正则表达式字面量中使用多个空格 /foo bar/

"no-restricted-modules": 0,//如果禁用了指定模块，使用就会报错

"no-return-assign": 1,//return 语句中不能有赋值表达式

"no-script-url": 0,//禁止使用javascript:void(0)

"no-self-compare": 2,//不能比较自身

"no-sequences": 0,//禁止使用逗号运算符

"no-shadow": 2,//外部作用域中的变量不能与它所包含的作用域中的变量或参数同名

"no-shadow-restricted-names": 2,//严格模式中规定的限制标识符不能作为声明时的变量名使用

"no-spaced-func": 2,//函数调用时 函数名与()之间不能有空格

"no-sparse-arrays": 2,//禁止稀疏数组， [1,,2]

"no-sync": 0,//nodejs 禁止同步方法

"no-ternary": 0,//禁止使用三目运算符

"no-trailing-spaces": 1,//一行结束后面不要有空格

"no-this-before-super": 0,//在调用super()之前不能使用this或super

"no-throw-literal": 2,//禁止抛出字面量错误 throw "error";

"no-undef": 1,//不能有未定义的变量

"no-undef-init": 2,//变量初始化时不能直接给它赋值为undefined

"no-undefined": 2,//不能使用undefined

"no-unexpected-multiline": 2,//避免多行表达式

"no-underscore-dangle": 1,//标识符不能以\_开头或结尾

"no-unneeded-ternary": 2,//禁止不必要的嵌套 var isYes = answer === 1 ? true : false;

"no-unreachable": 2,//不能有无法执行的代码

"no-unused-expressions": 2,//禁止无用的表达式

"no-unused-vars": [2, {"vars": "all", "args": "after-used"}],//不能有声明后未被使用的变量或参数

"no-use-before-define": 2,//未定义前不能使用

"no-useless-call": 2,//禁止不必要的call和apply

"no-void": 2,//禁用void操作符

"no-var": 0,//禁用var，用let和const代替

"no-warning-comments": [1, { "terms": ["todo", "fixme", "xxx"], "location": "start" }],//不能有警告备注

"no-with": 2,//禁用with

"array-bracket-spacing": [2, "never"],//是否允许非空数组里面有多余的空格

"arrow-parens": 0,//箭头函数用小括号括起来

"arrow-spacing": 0,//=>的前/后括号

"accessor-pairs": 0,//在对象中使用getter/setter

"block-scoped-var": 0,//块语句中使用var

"brace-style": [1, "1tbs"],//大括号风格

"callback-return": 1,//避免多次调用回调什么的

"camelcase": 2,//强制驼峰法命名

"comma-dangle": [2, "never"],//对象字面量项尾不能有逗号

"comma-spacing": 0,//逗号前后的空格

"comma-style": [2, "last"],//逗号风格，换行时在行首还是行尾

"complexity": [0, 11],//循环复杂度

"computed-property-spacing": [0, "never"],//是否允许计算后的键名什么的

"consistent-return": 0,//return 后面是否允许省略

"consistent-this": [2, "that"],//this别名

"constructor-super": 0,//非派生类不能调用super，派生类必须调用super

"curly": [2, "all"],//必须使用 if(){} 中的{}

"default-case": 2,//switch语句最后必须有default

"dot-location": 0,//对象访问符的位置，换行的时候在行首还是行尾

"dot-notation": [0, { "allowKeywords": true }],//避免不必要的方括号

"eol-last": 0,//文件以单一的换行符结束

"eqeqeq": 2,//必须使用全等

"func-names": 0,//函数表达式必须有名字

"func-style": [0, "declaration"],//函数风格，规定只能使用函数声明/函数表达式

"generator-star-spacing": 0,//生成器函数\*的前后空格

"guard-for-in": 0,//for in循环要用if语句过滤

"handle-callback-err": 0,//nodejs 处理错误

"id-length": 0,//变量名长度

"indent": [2, 4],//缩进风格

"init-declarations": 0,//声明时必须赋初值

"key-spacing": [0, { "beforeColon": false, "afterColon": true }],//对象字面量中冒号的前后空格

"lines-around-comment": 0,//行前/行后备注

"max-depth": [0, 4],//嵌套块深度

"max-len": [0, 80, 4],//字符串最大长度

"max-nested-callbacks": [0, 2],//回调嵌套深度

"max-params": [0, 3],//函数最多只能有3个参数

"max-statements": [0, 10],//函数内最多有几个声明

"new-cap": 2,//函数名首行大写必须使用new方式调用，首行小写必须用不带new方式调用

"new-parens": 2,//new时必须加小括号

"newline-after-var": 2,//变量声明后是否需要空一行

"object-curly-spacing": [0, "never"],//大括号内是否允许不必要的空格

"object-shorthand": 0,//强制对象字面量缩写语法

"one-var": 1,//连续声明

"operator-assignment": [0, "always"],//赋值运算符 += -=什么的

"operator-linebreak": [2, "after"],//换行时运算符在行尾还是行首

"padded-blocks": 0,//块语句内行首行尾是否要空行

"prefer-const": 0,//首选const

"prefer-spread": 0,//首选展开运算

"prefer-reflect": 0,//首选Reflect的方法

"quotes": [1, "single"],//引号类型 `` "" ''

"quote-props":[2, "always"],//对象字面量中的属性名是否强制双引号

"radix": 2,//parseInt必须指定第二个参数

"id-match": 0,//命名检测

"require-yield": 0,//生成器函数必须有yield

"semi": [2, "always"],//语句强制分号结尾

"semi-spacing": [0, {"before": false, "after": true}],//分号前后空格

"sort-vars": 0,//变量声明时排序

"space-after-keywords": [0, "always"],//关键字后面是否要空一格

"space-before-blocks": [0, "always"],//不以新行开始的块{前面要不要有空格

"space-before-function-paren": [0, "always"],//函数定义时括号前面要不要有空格

"space-in-parens": [0, "never"],//小括号里面要不要有空格

"space-infix-ops": 0,//中缀操作符周围要不要有空格

"space-return-throw-case": 2,//return throw case后面要不要加空格

"space-unary-ops": [0, { "words": true, "nonwords": false }],//一元运算符的前/后要不要加空格

"spaced-comment": 0,//注释风格不要有空格什么的

"strict": 2,//使用严格模式

"use-isnan": 2,//禁止比较时使用NaN，只能用isNaN()

"valid-jsdoc": 0,//jsdoc规则

"valid-typeof": 2,//必须使用合法的typeof的值

"vars-on-top": 2,//var必须放在作用域顶部

"wrap-iife": [2, "inside"],//立即执行函数表达式的小括号风格

"wrap-regex": 0,//正则表达式字面量用小括号包起来

"yoda": [2, "never"]//禁止尤达条件

}

}

## eslint忽略检查文件/文件夹

### 忽略文件.

整个文件范围内禁止规则出现警告

将/\* eslint-disable \*/放置于文件最顶部

/\* eslint-disable \*/

alert('foo');

在文件中临时禁止规则出现警告

将需要忽略的代码块用注释包裹起来

/\* eslint-disable \*/

alert('foo');

/\* eslint-enable \*/

对指定规则的启用或者禁用警告

将需要忽略的代码块用注释包裹起来

/\* eslint-disable no-alert, no-console \*/

alert('foo');

console.log('bar');

/\* eslint-enable no-alert, no-console \*/

对指定行禁用规则警告

此方法，有两种形式，参见下方。

alert('foo'); // eslint-disable-line

// eslint-disable-next-line

alert('foo');

在指定行上禁用指定的某个规则

alert('foo'); // eslint-disable-line no-alert

// eslint-disable-next-line no-alert

alert('foo');

在某个特定的行上禁用多个规则

alert('foo'); // eslint-disable-line no-alert, quotes, semi

// eslint-disable-next-line no-alert, quotes, semi

alert('foo');

### 忽略文件夹

在项目根目录建立.eslintignore文件，填上文件夹名就行，如min-app