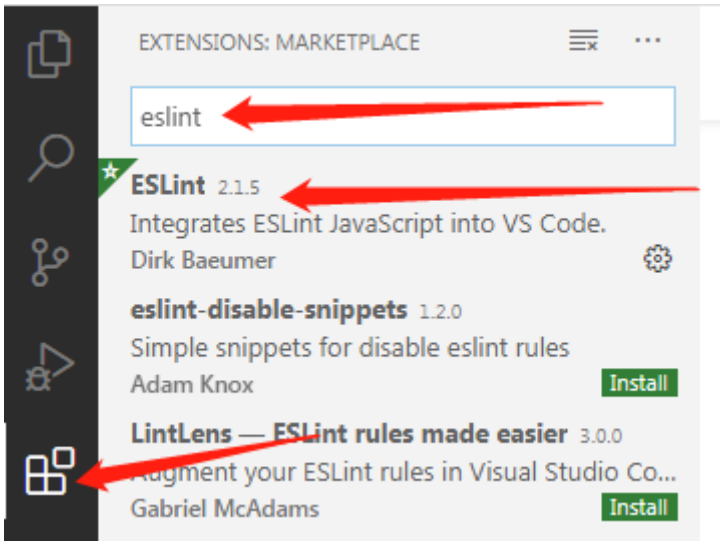
Visual Studio Code安装ESLint代码检测插件

概述

要养成一个好的编码习惯从自己编码开始，对自己代码的合理化命名，编码不仅对自己有好处，而且别人也容易读懂你的代码。目前公司提供插件化并且可配置的 JavaScript语法规则和代码风格的检查工具ESlint来规范代码，并且提供了eslint的配置文件，按照`.eslintrc.js`配置文件的规则来规范代码。

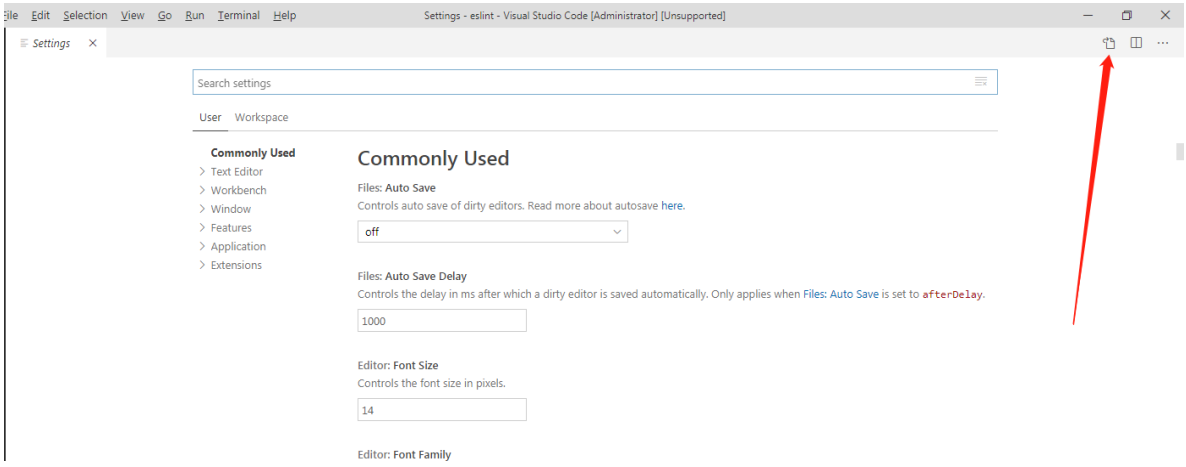
Visual Studio Code安装插件的步骤

1、打开vscode左侧扩展面板，搜索eslint，点击安装，重启vscode后生效



2、 安装好之后，还需要在vscode文件中进行设置：

通过 file --> preferences --> Settings 出现如下界面：



点击箭头所指的图标，将会打开`settings.json`文件，该文件为vscode配置文件

将下面代码复制到这个文件中（见附件）：

```
{
  // vscode默认启用了根据文件类型自动设置tabsize的选项
  "editor.detectIndentation": false,
  // 重新设定tabsize
  "editor.tabSize": 2,
  // #每次保存的时候自动格式化
  "editor.formatOnSave": true,
  // #每次保存的时候将代码按eslint格式进行修复
  "eslint.autoFixOnSave": true,
  // 添加 vue 支持
  "eslint.validate": [
    "javascript",
    "javascriptreact",
    {
      "language": "vue",
      "autoFix": true
    }
  ],
  //  #让prettier使用eslint的代码格式进行校验
  "prettier.eslintIntegration": true,
  //  #去掉代码结尾的分号
  "prettier.semi": false,
  //  #使用带引号替代双引号
  "prettier.singleQuote": true,
  //  #让函数(名)和后面的括号之间加个空格
  "javascript.format.insertSpaceBeforeFunctionParenthesis": true,
  // #让vue中的js按编辑器自带的ts格式进行格式化
  "vetur.format.defaultFormatter.js": "vscode-typescript",
  "vetur.format.defaultFormatterOptions": {
    "js-beautify-html": {
      "wrap_attributes": "force-aligned"
      // #vue组件中html代码格式化样式
    }
  },
  "window.zoomLevel": 0,
  "explorer.confirmDelete": false,
  "explorer.confirmDragAndDrop": false,
  "editor.renderControlCharacters": true,
  "editor.renderWhitespace": "all",
  "workbench.colorTheme": "Default Light+",
  "editor.codeActionsOnSave": {
    "source.fixAll.eslint": true
  }
}
```

3、在项目的`.eslintrc.js`中添加如下代码（见附件），如没有该文件则新建一个

```
module.exports = {
    'env': {
        'browser': true,
        'es6': true
    },
    'extends': [
```

```
        'eslint:recommended',
        'plugin:vue/essential'
    ],
    'globals': {
        'Atomics': 'readonly',
        'SharedArrayBuffer': 'readonly'
    },
    'parserOptions': {
        'ecmaVersion': 2018,
        'sourceType': 'module'
    },
    'plugins': [
        'vue'
    ],
// 规则的错误等级: 0, 1, 2
// 0或'off':   关闭规则
// 1或'warn':  开启规则, 作为警告输出, 退出码为0, 检查通过
// 2或'error': 开启规则, 作为错误输出, 退出码为1, 检查不通过
// ESLint官方规则: http://eslint.cn/docs/rules
    'rules': {
        "no-alert": 0,                    //禁止使用alert confirm prompt
        "no-array-constructor": 1,        //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": 1,             //禁止catch子句参数与外部作用域变量同名
        "no-class-assign": 1,             //禁止给类赋值
        "no-cond-assign": 1,              //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": 1,       //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": 1,            //禁止在正则表达式中使用控制字符
        "no-debugger": 1,                 //禁止使用debugger
        "no-delete-var": 1,               //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": 1,                //在创建对象字面量时不允许键重复 {a:1,a:1}
        "no-dupe-args": 1,                //函数参数不能重复
        "no-duplicate-case": 1,           //switch中的case标签不能重复
        "no-else-return": 1,              //如果if语句里面有return,后面不能跟else语句
        "no-empty": 1,                    //块语句中的内容不能为空
        "no-empty-character-class": 1,    //正则表达式中的[]内容不能为空
        "no-empty-label": 1,              //禁止使用空label
        "no-eq-null": 1,                  //禁止对null使用==或!=运算符
        "no-eval": 1,                     //禁止使用eval
        "no-ex-assign": 1,                //禁止给catch语句中的异常参数赋值
        "no-extend-native": 1,            //禁止扩展native对象
        "no-extra-bind": 1,               //禁止不必要的函数绑定
        "no-extra-boolean-cast": 1,       //禁止不必要的bool转换
        "no-extra-parens": 1,             //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": 1,             //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": 1,                 //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": 1,              //禁止不必要的嵌套块
        "no-lonely-if": 1,                //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": 1,                //字符串不能用\换行
        "no-multiple-empty-lines": [2, {"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": 1,               //禁止使用new Object()
        "no-new-require": 1,              //禁止使用new require
        "no-new-wrappers": 1,             //禁止使用new创建包装实例，new String new
Boolean new Number
        "no-obj-calls": 1,                //不能调用内置的全局对象，比如Math() JSON()
        "no-octal": 1,                    //禁止使用八进制数字
        "no-octal-escape": 1,             //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": 1,                    //禁止使用__proto__属性
        "no-redeclare": 2,                //禁止重复声明变量
        "no-regex-spaces": 1,             //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
```

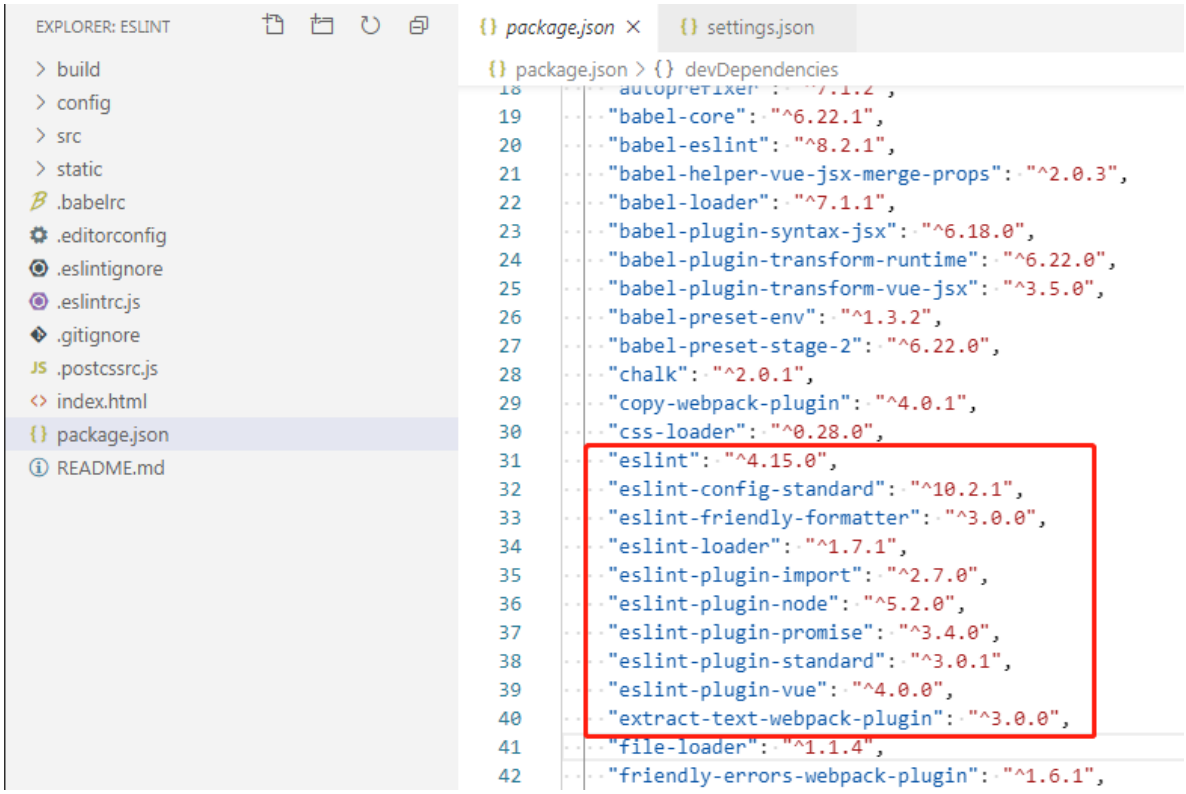
```
        "no-undef-init": 2,                 //变量初始化时不能直接给它赋值为undefined
        "no-undefined": 1,                  //不能使用undefined
        "no-unexpected-multiline": 1,       //避免多行表达式
        "no-underscore-dangle": 1,          //标识符不能以_开头或结尾
        "no-unneeded-ternary": 2,           //禁止不必要的嵌套 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": 1,               //禁止不必要的call和apply
        "no-void": 1,                       //禁用void操作符
        "no-var": 2,                        //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
        "no-whitespace-before-property": 2,     //属性前面不能加空格
        "array-bracket-spacing": [2, "never"],  //是否允许非空数组里面有多余的空格
        "arrow-parens": 2,                  //箭头函数用小括号括起来
        "arrow-spacing": 2,                 //=>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
        "block-spacing": [2, "always"],     //代码块首尾留空格
        "brace-style": [2, "allman"],       //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": [2, {"before": false, "after": true}], //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
        "curly": [2, "multi-line"],         //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": [2, "^.*(e|E)rr"],        //强制回调错误处理
        "id-length": [2, {"max": 20}],   //变量名长度
        "id-match": [2, "^[a-z$]+([a-zA-Z0-9]+)*$"],     //命名检测
        "indent": [2, 2, {"SwitchCase": 2}],         //缩进2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": 1,                       //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": 1,             //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": 1,                         //parseInt必须指定第二个参数
        "require-yield": 0,                 //生成器函数必须有yield
        "semi": [2, "always"],              //语句强制分号结尾 always/never
        "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": [2, "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": [2, "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": [2, {"int32Hint": false}],     ////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": [2, "always"], //注释首尾留空格
        "strict": 1,                        //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": 1,                   //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
    }
}
```

4、安装自动修复插件

在`package.json`添加eslint安装模块

```
"eslint": "^4.15.0"
"eslint-config-standard": "^10.2.1"
"eslint-friendly-formatter": "^3.0.0"
"eslint-loader": "^1.7.1"
"eslint-plugin-import": "^2.7.0"
"eslint-plugin-node": "^5.2.0"
"eslint-plugin-promise": "^3.4.0"
"eslint-plugin-standard": "^3.0.1"
"eslint-plugin-vue": "^4.0.0"
"extract-text-webpack-plugin": "^3.0.0"
```



使用插件去规范代码

安装好了，当我们编写不符合eslint规范的代码时，启动项目会报错，比如

当我们编写代码的时候，ctrl + s保存代码后，便会自动修复格式不正确的代码

附件1：`settings.json`配置文件

附件2：`.eslintrc.js`配置文件