



Vue CLI

🔧 Vue.js 开发的标准工具

1:什么是Vue-CLI

1. Vue-CLI 是 Vue 官方提供的, 用来搭建项目脚手架的工具, 它是 Vue.js 开发的标准工具, 它已经集成了 webpack , 内置好了很多常用配置, 使得我们在使用 Vue 开发项目时更加的标准化。
注意新版本是对应 `@vue/cli` , 老版本的 `vue-cli` 两个版本差异较大
2. 作用: 通过 Vue-CLI 下载模板项目。
3. 官方文档: <https://cli.vuejs.org/zh/>
4. github站点: <https://github.com/vuejs/vue-cli>

2:安装Vue-CLI

全局安装Vue-CLI

Win:

命令: `npm i -g @vue/cli@3.10.0`

OS:

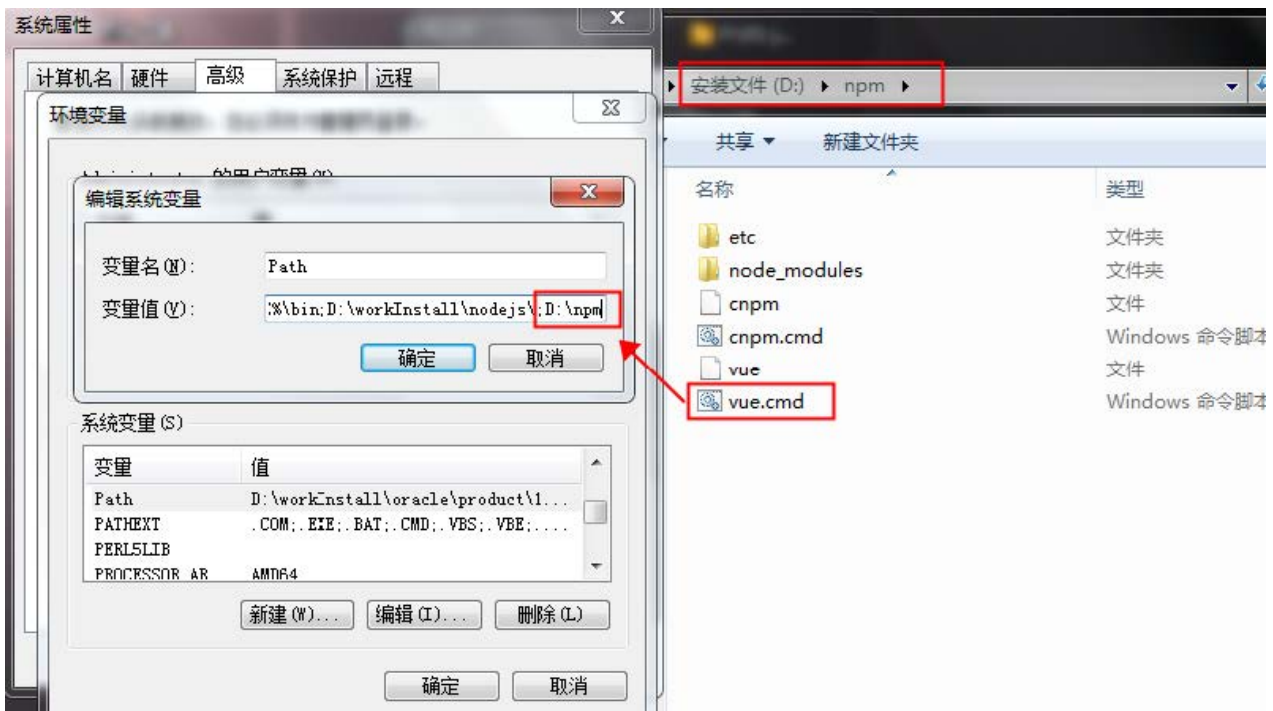
命令: `sudo npm i -g --unsafe-perm @vue/cli@3.10.0`

安装之后查看版本

命令: `vue -V`

注意:

- 如果出现“vue不是内部或者外部命令”
- 解决方案，配置环境变量
- 首先查看 全局安装目录 命令 `npm root -g`，在我的电脑进入全局安装目录下，找到 vue.cmd
- 右键计算机，属性=>高级系统设置=>环境变量，将 vue.cmd 的路径加入环境变量，点击“确定”



重启命令行窗口，输入 `vue -V`

3:使用脚手架创建项目

Win:

命令: `vue create 项目名称`

注意项目名称必须小写

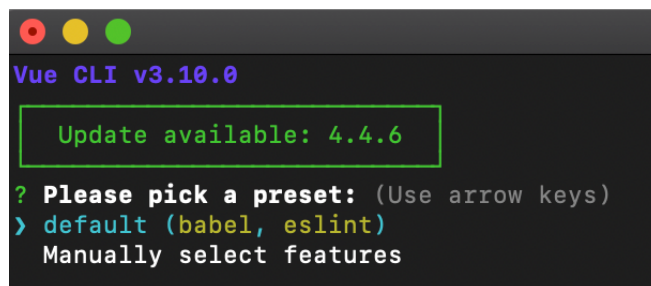
Os:

命令: `sudo vue create 项目名称`

如果出现权限错误:

```
1 //会报权限错误, 先不管
2 vue create myproject1
3
4 //进入项目文件夹
5 cd myproject1
6
7 //根据之前的配置再次安装, 不过这次加上参数
8 npm install --production --unsafe-perm=true --allow-root
9
10 //安装成功, 测试运行正常
11 npm run serve
12
13 //虽然上面这样运行正常了, 不过接着在用vue add xxx 命令的时候又是报权限错误, 这样没法再用上面的办法解决了。
14 sudo npm i -g npm
15 //给/Users/stormer/.npm这个文件夹读写权限,
16 sudo chmod -R 777 /Users/stormer/.npm
17 //再测试npm和vue命令都可以正常运行了
```

• 3.1:创建默认项目



会提示选择默认 (default) 还是手动 (Manually),默认的配置非常适合快速创建一个新项目的原型,而手动设置则提供了更多的选项。这里选择 default , 等待一会 (如有提示等待, 一直回车执行下去就行) 。

• 3.2:自定义创建项目

```
Vue CLI v3.10.0

Update available: 4.4.6

? Please pick a preset:
  default (babel, eslint)
> Manually select features
```

- 选择安装的模块

```
Vue CLI v3.10.0

Update available: 4.4.6

? Please pick a preset: Manually select features
? Check the features needed for your project:
  ● Babel
  ○ TypeScript
  ○ Progressive Web App (PWA) Support
  ● Router
  ○ Vuex
> ● CSS Pre-processors
  ● Linter / Formatter
  ○ Unit Testing
  ○ E2E Testing
```

“

Babel: (常用) 解决兼容性问题,支持ES6的代码转译成浏览器能识别的代码

TypeScript:是一种给JavaScript添加特性的语言扩展,增加了很多功能,微软开发的

Progressive Web App (PWA) Support:渐进式的Web应用程序支持

Router: (常用) 是 vue-router 路由。

Vuex:是Vue.js应用程序的状态管理模式+库 (常用)。

CSS Pre-processors: (常用) 支持 CSS 预处理器, Sass/Less预处理器。

Linter / Formatter: (常用) 支持代码风格检查和格式化。

Unit Testing:支持单元测试。

E2E Testing:支持 E2E 测试。

```
demo — node /usr/local/bin/vue create demo1 — 144x45
Vue CLI v3.10.0

Update available: 4.4.6

? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, CSS Pre-processors, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) (Y/n)
```

是否使用 history 模式路由, 点击回车即可

```
demo — node /usr/local/bin/vue create demo1 — 144x45
Vue CLI v3.10.0
Update available: 4.4.6
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, CSS Pre-processors, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): (Use arrow keys)
> Sass/SCSS (with dart-sass)
  Sass/SCSS (with node-sass)
  Less
  Stylus
```

选择CSS预处理器

```
demo — node /usr/local/bin/vue create demo1 — 144x45
Vue CLI v3.10.0
Update available: 4.4.6
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, CSS Pre-processors, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with dart-sass)
? Pick a linter / formatter config:
  ESLint with error prevention only
  ESLint + Airbnb config
  ESLint + Standard config
> ESLint + Prettier
```

检查和格式化代码

```
demo — node /usr/local/bin/vue create demo1 — 144x45
Vue CLI v3.10.0
Update available: 4.4.6
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, CSS Pre-processors, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with dart-sass)
? Pick a linter / formatter config: Prettier
? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection)
> ● Lint on save
  ○ Lint and fix on commit
```

保存就检测

```
demo — node /usr/local/bin/vue create demo1 — 144x45
Vue CLI v3.10.0
Update available: 4.4.6
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router, CSS Pre-processors, Linter
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported by default): Sass/SCSS (with dart-sass)
? Pick a linter / formatter config: Prettier
? Pick additional lint features: (Press <space> to select, <a> to toggle all, <i> to invert selection) Lint on save
? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? (Use arrow keys)
> In dedicated config files
  In package.json
```

把 babel、ESLint、Csspost 这些配置文件如何保存，选择以独立文件保存，另一个选项为保存在 package.json 中，默认则是保存在 package.json 中

New **minor** version of npm available! 6.4.1 → 6.14.6
Changelog: <https://github.com/npm/cli/releases/tag/v6.14.6>
Run **npm install -g npm** to update!

Running completion hooks...

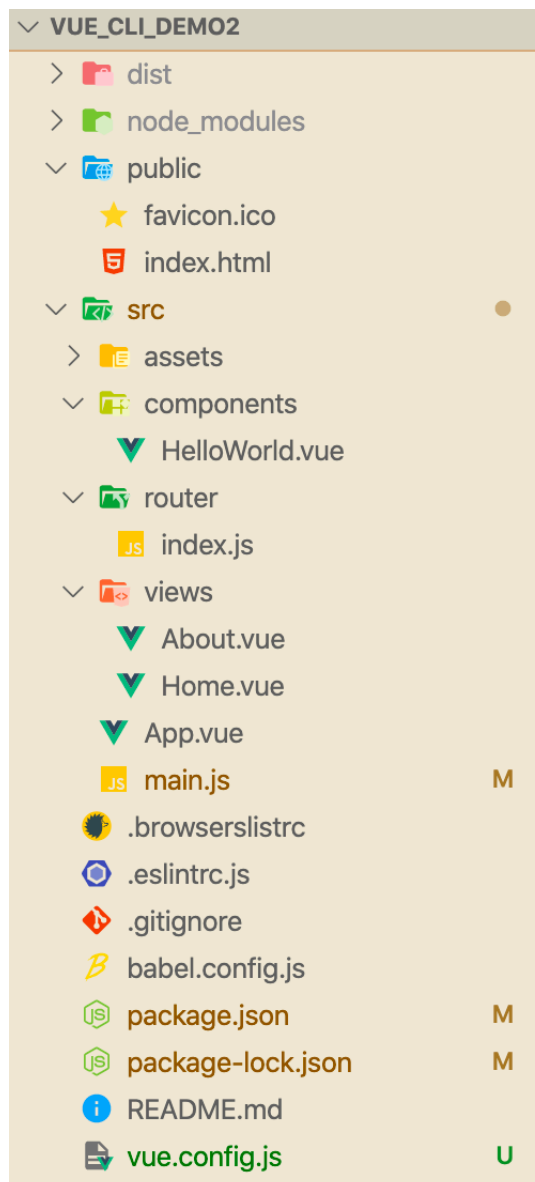
Generating README.md...

Successfully created project **demo1**.
Get started with the following commands:

```
$ cd demo1  
$ npm run serve
```

安装完毕后 切换到项目目录,
执行 **npm run serve** 打开工程

- 目录结构



- 各模块功能

“

工程名（不能有大写字母）

|-- node_modules: 存放下载依赖的文件夹

|-- public: 存放不会变动静态的文件，它与src/assets的区别在于，public目录中的文件不被webpack打包处理，会原

样拷贝到dist目录下

|-- index.html: 主页面文件

|-- favicon.ico: 在浏览器上显示的图标

|-- src: 源码文件夹

|-- assets: 存放组件中的静态资源

|-- components: 存放一些公共组件

|-- views: 存放所有的路由组件

|-- App.vue: 应用根主组件

|-- main.js: 应用入口 js

|-- .browserslistrc: 指定了项目可兼容的目标浏览器范围, 对应是package.json 的 browserslist选项

|-- .eslintrc.js: eslint相关配置

|-- .gitignore: git 版本管制忽略的配置

|-- babel.config.js: babel 的配置,即ES6语法编译配置

|-- package-lock.json: 用于记录当前状态下实际安装的各个包的具体来源和版本号等, 保证其他人在 npm install 项

目时大家的依赖能保证一致.

|-- package.json: 项目基本信息,包依赖配置信息等

|-- postcss.config.js: postcss一种对css编译的工具，类似babel对js的处理

|-- README.md: 项目描述说明的 readme 文件

3.3:Vue-CLI基本指令

CLI 服务 (@vue/cli-service) 是一个开发环境依赖。针对绝大部分应用优化过的内部的 webpack 配置；在一个 Vue CLI 项目中， @vue/cli-service 模块安装了一个名为 vue-cli-service 的命令。

```
"scripts": {  
  "serve": "vue-cli-service serve",  
  "build": "vue-cli-service build",  
  "lint": "vue-cli-service lint"  
},
```

在 package.json 中的 scripts 指定 vue-cli-service 相关命令。

- serve 启动一个开发环境服务器(基于 webpack-dev-server)修改组件代码后, 会自动热模块替换
- build 会项目根目录下自动创建一个 dist/ 目录, 打包后的文件都在其中, js文件: 是经过压缩加密的, 如果运行时报错, 输出的错误信息无法准确定位到哪里的代码报错。map文件: 文件比较大, 代码未加密, 可以准确的输出是哪一行哪一列有错。
- lint 使用 Eslint 进行检查并修复代码的规范。比如: 将 main.js 中的格式多加个空格, 执行 npm run lint 后它会自动的去掉多余空格。

3.4:.browserslistrc配置浏览器兼容性

“

last 2 versions 每一个主流浏览器的最后5个版本

last 2 Chrome versions 谷歌浏览器的最后两个版本

> 2% 市场占有率大于2%

> 2% in US 美国市场占有率大于2%

ie 6-8 ie浏览器6-8

Firefox > 20 火狐版本>20

Firefox < 20 火狐<20

since 2013 2013年之后发布的所有版本

iOS 7 指定IOS 7浏览器

3.5:自定义配置

在项目根目录下创建 vue.config.js 文件


```

1  module.exports = {
2    devServer: {
3      port: 8001, // 端口号, 如果端口被占用, 会自动提升 1
4      open: true, // 启动服务自动打开浏览器
5      https: false, // 协议
6      host: "localhost", // 主机名, 也可以 127.0.0.1 或 做真机测试时候 0.0.0.0
7    },
8    lintOnSave: false, // 是否关闭eslint语法检查, 默认 true, 警告仅仅会被输出到命令
    行, 且不会使得编译失败。
9    outputDir: "dist", // 默认是 dist ,存放打包文件的目录,
10   assetsDir: "assets", // 存放生成的静态资源 (js、css、img、fonts) 的 (相对于
    outputDir) 目录
11   indexPath: "out/index.html", // 默认 index.html, 指定生成的 index.html 的输
    出路径 (相对于 outputDir)。
12   productionSourceMap: false, // 打包时, 不生成 .map 文件, 加快打包构建
13 }

```

3.6:配置ESLint语法检查

“

ESLint 是一个语法规则和代码风格的检查工具, 可以用来保证写出语法正确、风格统一的代码。如果我们开启了 Eslint, 也就意味着要接受它 非常苛刻的语法检查, 包括空格不能少些或多些, 必须单引不能双引, 语句后不可以写分号等等, 这些规则其实是可以设置的, 作为前端的初学者, 最好先关闭这种校验, 否则会浪费很多精力在语法的规范性上。如果以后在工作中, 建议开启。Vue-Cli 中默认是开启的

```

1  // https://eslint.org/docs/user-guide/configuring
2  module.exports = {
3    root: true,
4    parserOptions: {
5      parser: 'babel-eslint',
6      sourceType: 'module'
7    },
8    env: {
9      browser: true
10   },
11   extends: [
12     'plugin:vue/base'
13   ],
14   // required to lint *.vue files
15   plugins: [

```

```

16     'vue'
17 ],
18 // add your custom rules here
19 'rules': {
20     // allow paren-less arrow functions
21     'indent': [2, 2], // 两个空格的缩进
22     'quotes': [2, 'single'], // js必须使用单引号
23     "linebreak-style": [0, "error", "windows"], //允许windows开发环境
24     // 'semi': [2, 'always'], // 语句强制分号结尾
25     'no-console': [1], // 不允许console语句
26     'no-unused-vars': [1], // 声明了变量但是没有使用检测
27     'space-unary-ops': [1, { 'words': true, 'nonwords': false }], // 一元运
    算符的前/后要不要加空格
28     'brace-style': [2, '1tbs', { 'allowSingleLine': false }], // 大括号风格
29     'comma-spacing': [2, { 'before': false, 'after': true }], // 逗号后有
    空格, 前没有空格
30     'comma-style': [2, 'last'], // 逗号跟在结尾
31     'key-spacing': [2, { 'beforeColon': false, 'afterColon': true }], // 对
    象字面量中冒号的前后空格
32     'lines-around-comment': [ // 行前/行后备注
33         2, {
34             'beforeBlockComment': false, // 段注释的前后
35             'beforeLineComment': false, // 行注释的前面
36             'afterBlockComment': false, // 块注释的后面
37             'afterLineComment': false, // 行注释的后面
38             'allowBlockStart': true,
39             'allowObjectStart': true,
40             'allowArrayStart': true
41         }],
42     'max-depth': [2, 4], // 代码最多允许4层嵌套
43     'max-len': [1, 160, 2],
44     'max-nested-callbacks': [2, 3], // 回调嵌套深度
45     'max-params': [2, 5], // 函数最多只能有5个参数
46     'max-statements': [1, 80], // 单个函数最多80条语句
47     'no-array-constructor': [2], // 禁止使用数组构造器
48     'no-lonely-if': 2, // 禁止else语句内只有if语句
49     'no-multiple-empty-lines': [2, { 'max': 3, 'maxEOF': 1 }], // 空行最多不
    能超过2行
50     'no-nested-ternary': 2, // 不使用嵌套的三元表达式
51     'no-spaced-func': 2, // 函数调用时 函数名与()之间不能有空格
52     'no-trailing-spaces': 2, // 一行结束后面不要有空格
53     'no-unneeded-ternary': 2, // 禁止不必要的嵌套 var isYes = answer === 1 ?
    true : false;简单的判断用三元表达式代替
54     'object-curly-spacing': [2, 'always', { // 大括号内是否允许不必要的空格
    always始终允许; never始终不允许
55         'objectsInObjects': false,
56         'arraysInObjects': false

```

```

57     }],
58     'arrow-spacing': 2, // =>的前/后括号
59     'block-scoped-var': 2, // 块语句中使用var
60     'no-dupe-class-members': 2,
61     // 'no-var': 1, // 禁用var, 用let和const代替
62     'object-shorthand': [1, 'always'], // 强制对象字面量缩写语法
63     'array-bracket-spacing': [2, 'never'], // 是否允许非空数组里面有多余的空格
64     'operator-linebreak': [2, 'after'], // 换行时运算符在行尾还是行首
65     'semi-spacing': [2, { 'before': false, 'after': true }], // 分号前后空格
66     'keyword-spacing': ['error'],
67     'space-before-blocks': 2, // 不以新行开始的块{前面要不要有空格
68     'block-spacing': [2, 'always'],
69     'space-before-function-paren': [2, 'never'], // 函数定义时括号前面要不要有
    空格
70     'space-in-parens': [2, 'never'], // 小括号里面要不要有空格
71     'spaced-comment': [1, 'always',
72       { 'exceptions': ['-', '*', '+']
73       }], // 注释风格要不要有空格什么的
74     'arrow-parens': 0,
75     // allow async-await
76     'generator-star-spacing': 0,
77     // allow debugger during development
78     'no-debugger': process.env.NODE_ENV === 'production' ? 2 : 0
79   },
80   globals: {
81     '$': false,
82     'jquery': false,
83     'ActiveXObject': false,
84     'arbor': true,
85     'layer': false
86   }
87 };

```

3.7:发布工程

1. 在项目的根目录下新建 vue.config.js 文件（是根目录，不是src目录）

```

1   module.exports = {
2     publicPath: '/工程名/'
3   }

```

2. 构建打包 npm run build

3. dist 目录拷贝到 tomcat\webapps 目录下，然后将 dist 目录名 修改为 工程名
4. 打开 tomcat/bin/startup.bat 启动服务器（注意之前启动的 8080 端口项目要先停止）
5. 访问：<http://localhost:8080/> 工程名/

3.8:VSCode前端工程字母含义

“

- A: 新增的文件.
- C: 文件的一个新拷贝.
- D: 删除的一个文件.
- M: 文件的内容或者mode被修改了.
- R: 文件名被修改了。
- T: 文件的类型被修改了。
- U: 文件没有被合并(你需要完成合并才能进行提交)
- X: 未知状态

3.9:各组件版本号对照

package.json X

package.json > ...

```
1  {
2    "name": "vue_cli_demo2",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "serve": "vue-cli-service serve",
7      "build": "vue-cli-service build",
8      "lint": "vue-cli-service lint"
9    },
10   "dependencies": {
11     "axios": "^0.19.2",
12     "core-js": "^3.6.5",
13     "element-ui": "^2.13.2",
14     "vue": "^2.6.11",
15     "vue-router": "^3.2.0"
16   },
17   "devDependencies": {
18     "@vue/cli-plugin-babel": "^4.4.0",
19     "@vue/cli-plugin-eslint": "^4.4.0",
20     "@vue/cli-service": "^4.4.0",
21     "@vue/eslint-config-prettier": "^6.0.0",
22     "babel-eslint": "^10.1.0",
23     "eslint": "^6.7.2",
24     "eslint-plugin-prettier": "^3.1.3",
25     "eslint-plugin-vue": "^6.2.2",
26     "prettier": "^1.19.1",
27     "sass": "^1.26.5",
28     "sass-loader": "^8.0.2",
29     "vue-template-compiler": "^2.6.11"
30   }
31 }
```

