

Vue CLI相关

王红元

微博: coderwhy

微信: 372623326



实力IT教育 www.520it.com

什么是Vue CLI

- 如果你只是简单写几个Vue的Demo程序, 那么你不需要Vue CLI.
- 如果你在开发大型项目, 那么你需要, 并且必然需要使用Vue CLI
 - 使用Vue.js开发大型应用时, 我们需要考虑代码目录结构、项目结构和部署、热加载、代码单元测试等事情。
 - 如果每个项目都要手动完成这些工作, 那无以效率比较低效, 所以通常我们会使用一些脚手架工具来帮助完成这些事情。
- CLI是什么意思?
 - CLI是Command-Line Interface, 翻译为命令行界面, 但是俗称脚手架.
 - Vue CLI是一个官方发布 vue.js 项目脚手架
 - 使用 vue-cli 可以快速搭建Vue开发环境以及对应的webpack配置.
- 脚手架长什么样子?



Vue CLI使用前提 - Node

■ 安装NodeJS

- 可以直接在官方网站中下载安装.
- 网址: <http://nodejs.cn/download/>

■ 检测安装的版本

- 默认情况下自动安装Node和NPM
- Node环境要求8.9以上或者更高版本

```
[bogon:Vue-TabBar xmg$ node -v  
v10.5.0  
[bogon:Vue-TabBar xmg$ npm -v  
6.1.0
```

■ 什么是NPM呢?

- NPM的全称是Node Package Manager
- 是一个NodeJS包管理和分发工具, 已经成为了非官方的发布Node模块(包)的标准。
- 后续我们会经常使用NPM来安装一些开发过程中依赖包.

cnpm安装

由于国内直接使用 npm 的官方镜像是非常慢的, 这里推荐使用淘宝 NPM 镜像。

你可以使用淘宝定制的 cnpm (gzip 压缩支持) 命令行工具代替默认的 npm:

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

这样就可以使用 cnpm 命令来安装模块了:

```
cnpm install [name]
```

Vue CLI使用前提 - Webpack

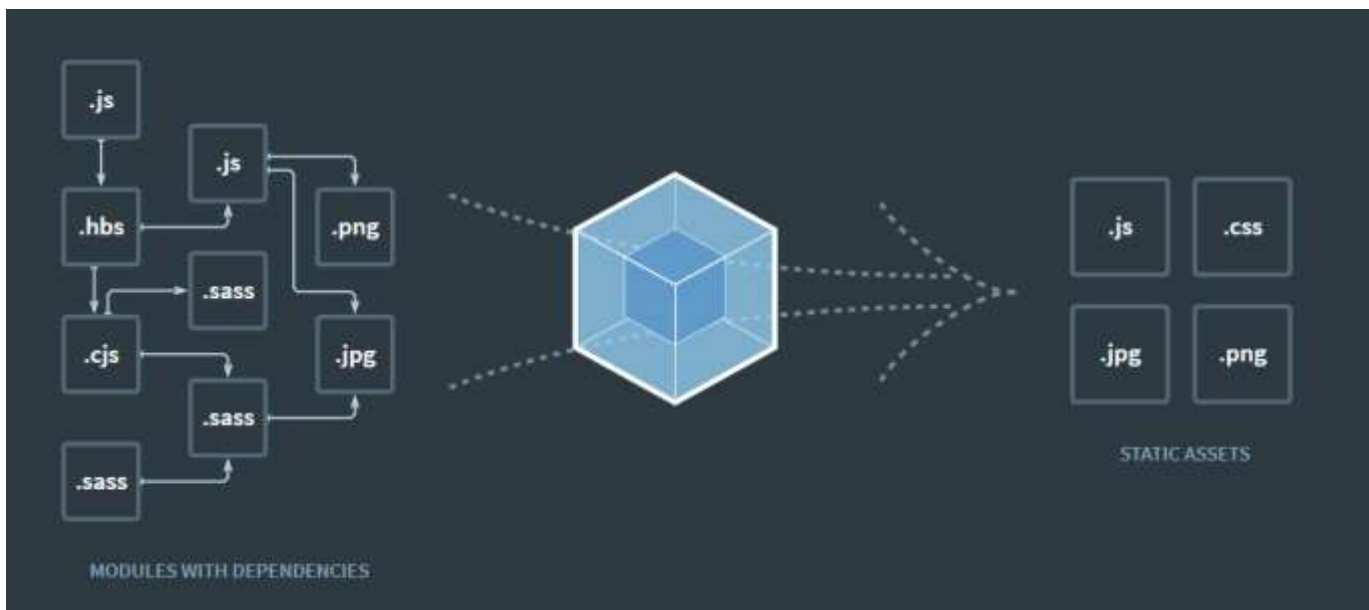
■ Vue.js官方脚手架工具就使用了webpack模板

- 对所有的资源会压缩等优化操作

- 它在开发过程中提供了一套完整的功能，能够使得我们开发过程中变得高效。

■ Webpack的全局安装

- `npm install webpack -g`



Vue CLI的使用

■ 安装Vue脚手架

□ `npm install -g @vue/cli`

```
bogon:~ xmg$ vue --version
3.0.4
```

■ 注意：上面安装的是Vue CLI3的版本，如果需要想按照Vue CLI2的方式初始化项目时不可以的。

拉取 2.x 模板 (旧版本)

Vue CLI 3 和旧版使用了相同的 `vue` 命令，所以 Vue CLI 2 (`vue-cli`) 被覆盖了。如果你仍然需要使用旧版本的 `vue init` 功能，你可以全局安装一个桥接工具：

```
npm install -g @vue/cli-init
# `vue init` 的运行效果将会跟 `vue-cli@2.x` 相同
vue init webpack my-project
```

■ Vue CLI2初始化项目

□ `vue init webpack my-project`

■ Vue CLI3初始化项目

□ `vue create my-project`

Vue CLI2详解

1.会根据这个名称创建一个文件夹，存放之后项目的内容
该名称也会作为默认的项目名称，但是不能包含大写字母等

2.项目名称，不能包含大写

3.作者的信息，会默认从git中读取信息

4.后面详细介绍

5.vue-router，这里我选择no，后面自己安装

6.ESLint检测代码规范，看自己的情况

7.单元测试
某些公司强制要求写单元测试

8.e2e测试，end to end
安装Nightwatch，是一个利用selenium或
webdriver或phantomjs等进行自动化测试的框架

9.选择用yarn或者npm安装都可以

```
[bogon:Vue xmg$ vue init webpack myvuejsproject]
[?] Project name myvuejsproject
[?] Project description A Vue.js project
[?] Author coderwhy <372623326@qq.com>
[?] Vue build runtime
[?] Install vue-router? No
[?] Use ESLint to lint your code? No
[?] Set up unit tests No
[?] Setup e2e tests with Nightwatch? No
[?] Should we run `npm install` for you after the project has been created? (recom
mended) (Use arrow keys)
> Yes, use NPM
  Yes, use Yarn
  No, I will handle that myself
```

mode by coderwhy
微博:coderwhy

目录结构详解



Runtime-Compiler和Runtime-only的区别

> Runtime + Compiler: recommended for most users

Runtime-only: about 6KB lighter min+gzip, but templates (or any Vue-specific HTML) are ONLY allowed in .vue files - render functions are required elsewhere

■ 简单总结

- 如果在之后的开发中，你依然使用template，就需要选择Runtime-Compiler
- 如果你之后的开发中，使用的是.vue文件开发，那么可以选择Runtime-only

运行时+编译器(Runtime + Compiler)版本 vs 只含有运行时版本(Runtime-only)

如果你需要在客户端编译模板（例如，向 `template` 选项传入一个字符串，或者需要将模板中的非 DOM 的 HTML 挂载到一个元素），你需要带有编译器的版本，因而需要完整构建版本。

```
// 这种情况需要编译器(compiler)
new Vue({
  template: '<div>{{ hi }}</div>'
})

// 这种情况不需要
new Vue({
  render (h) {
    return h('div', this.hi)
  }
})
```

在使用 `vue-loader` 或 `vueify` 时，`*.vue` 文件中的模板会在构建时(build time)预编译(pre-compile)为 JavaScript。最终生成的 bundle 中你不再需要编译器(compiler)，因此可以直接使用只含有运行时的构建版本(runtime-only)。

由于只含有运行时构建版本(runtime-only)比完整构建版本(full-build)轻量大约 30%，你应该尽可能使用只含有运行时的构建版本。如果你还是希望使用完整构建版本，则需要在打包器中配置别名：

由于运行时版本的构建比其全面版本的重量轻约30%，因此你可以随时使用它。如果你仍然希望使用完整版本，则需要在捆绑程序中配置别名：

render和template

■ Runtime-Compiler 和 Runtime-only

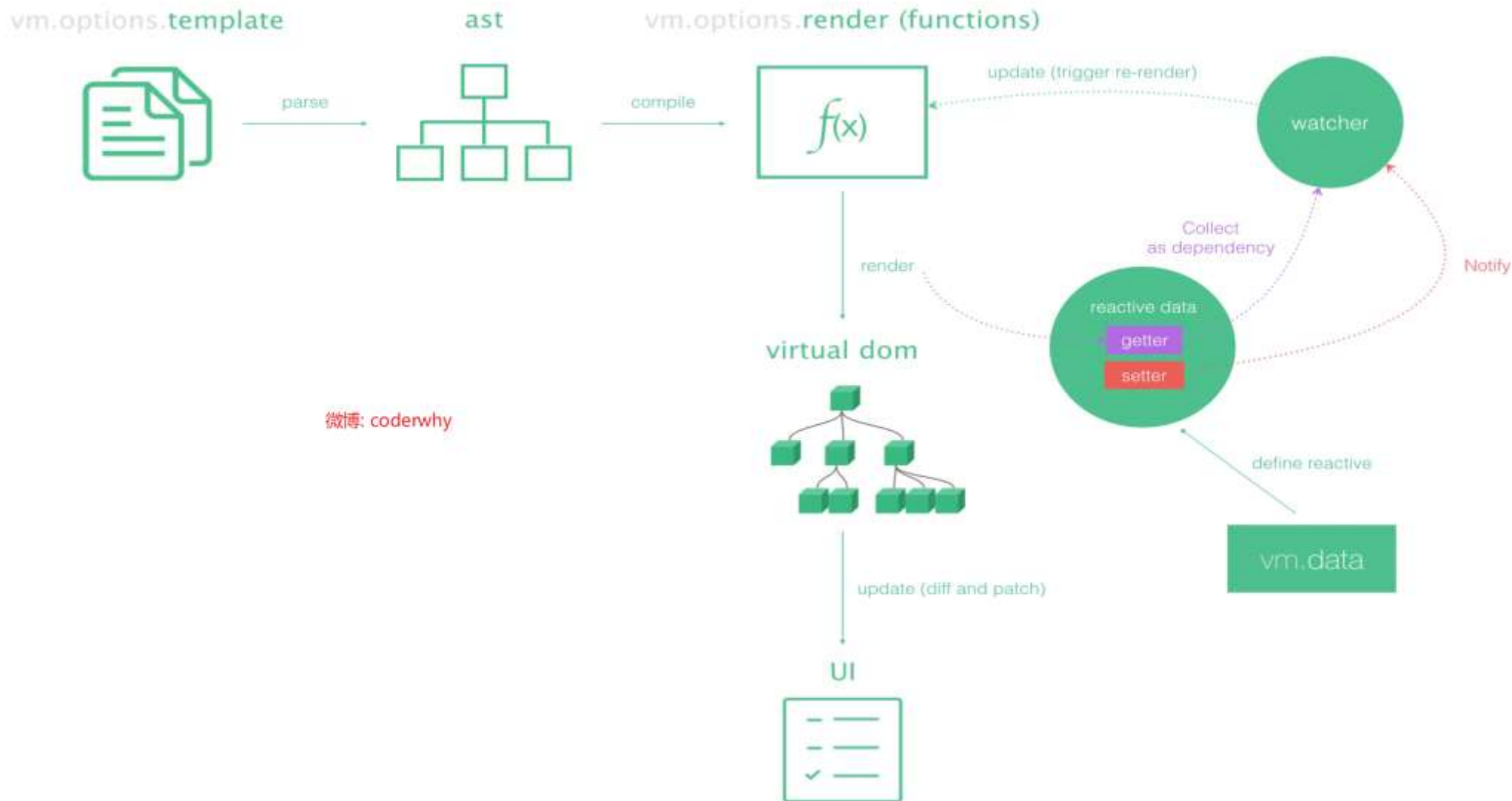
```
new Vue({  
  el: '#app',  
  components: { App },  
  template: '<App/>'  
})
```

```
new Vue({  
  el: '#app',  
  render: h => h(App)  
})
```

■ 为什么存在这样的差异呢？

- 我们需要先理解Vue应用程序是如何运行起来的。
- Vue中的模板如何最终渲染成真实DOM。
- 我们来看下面的一幅图。

Vue程序运行过程



render函数的使用

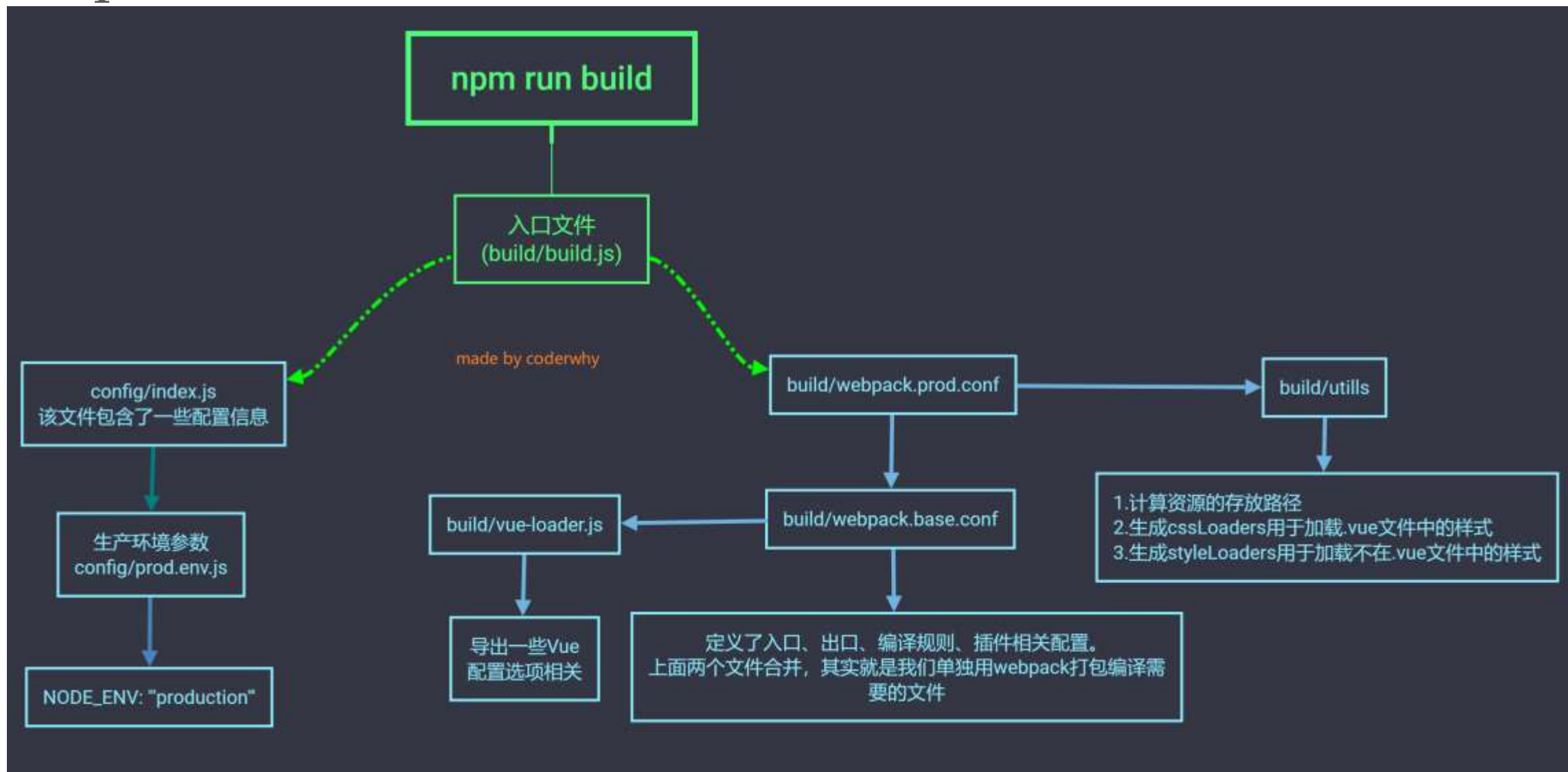
```
new Vue({
  el: '#app',
  render: (createElement) => {
    // 1.使用方式一:
    return createElement('标签', '相关数据对象(可以不传)', ['内容数组'])
    // 1.1.render函数基本使用
    return createElement('div', {class: 'box'}, ['coderwhy'])
    // 1.2.嵌套render函数
    return createElement('div', {class: 'box'}, ['coderwhy', createElement('h2', ['标题啊'])])
  }
})
```

```
const cpn = Vue.component('cpn', {
  template: '<div>我是cpn组件</div>',
  data() {
    return {
    }
  }
})
```

```
new Vue({
  el: '#app',
  render: (createElement) => {
    // 2.使用方式二: 传入一个组件对象
    return createElement(cpn)
  }
})
```

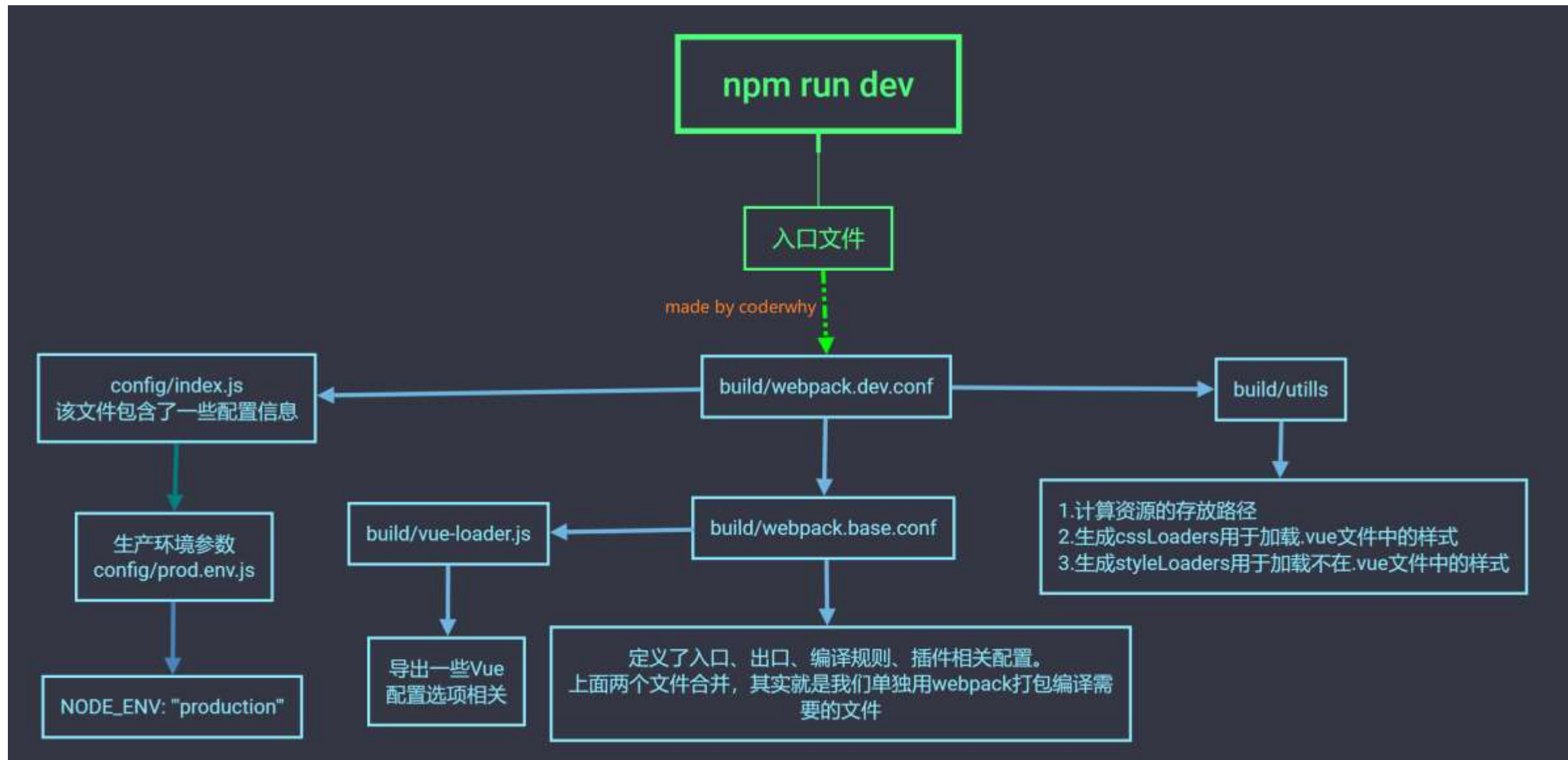


npm run build





npm run dev



修改配置: webpack.base.conf.js起别名

```
resolve: {  
  extensions: ['.js', '.vue', '.json'],  
  alias: {  
    '@': resolve('src'),  
    'pages': resolve('src/pages'),  
    'common': resolve('src/common'),  
    'components': resolve('src/components'),  
    'network': resolve('src/network')  
  }  
},
```


认识Vue CLI3

■ 今年8月份刚刚发布



■ vue-cli 3 与 2 版本有很大区别

- ❑ vue-cli 3 是基于 webpack 4 打造, vue-cli 2 还是 webapck 3
- ❑ vue-cli 3 的设计原则是 “0配置”, 移除的配置文件根目录下的, build和config等目录
- ❑ vue-cli 3 提供了 vue ui 命令, 提供了可视化配置, 更加人性化
- ❑ 移除了static文件夹, 新增了public文件夹, 并且index.html移动到public中

Vue CLI v3.0.4

? Please pick a preset: (Use arrow keys)

> coderwhy (babel) 1.选择配置方式
default (babel, eslint)
Manually select features

? Check the features needed for your project: (Press <space> to toggle all, <i> to invert selection)

> ● Babel
○ TypeScript
○ Progressive Web App (PWA) Support
○ Router
○ Vuex
○ CSS Pre-processors
● Linter / Formatter
○ Unit Testing
○ E2E Testing

2.选择自己需要的配置
空格选中或反选

made by coderwhy
微博: coderwhy

? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? (Use arrow keys)

> In dedicated config files 3.对应的配置单独生成文件还是放在package.json
In package.json

? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.? (Use arrow keys)

? Save this as a preset for future projects? (y/N) ☐
4.要不要把刚才自己选择配置保存下来

? Save this as a preset for future projects? Yes

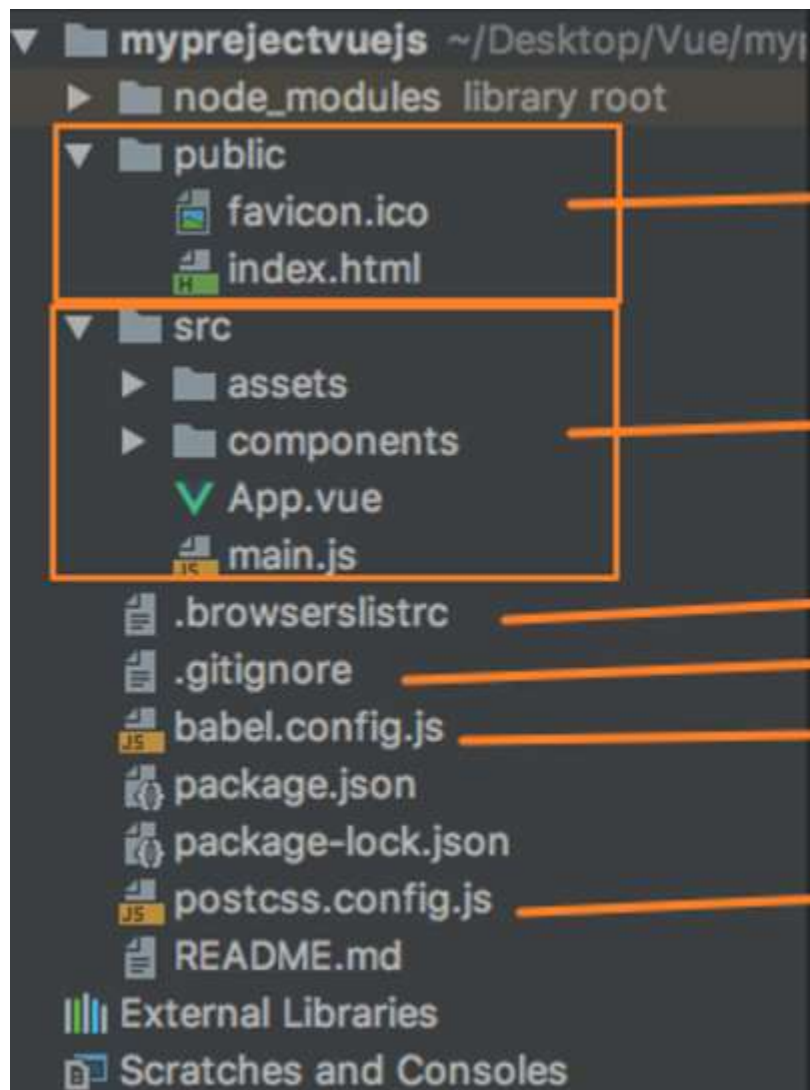
? Save preset as: mypreset

5.设置保存的名称

Vue CLI v3.0.4

🌟 Creating project in /Users/xmg/Desktop/Vue/myprojectvuejs.
📁 Initializing git repository...
⦿ Installing CLI plugins. This might take a while...

目录结构详解



相当于CLI2中的static目录

源代码，以后就在这里写

浏览器相关支持情况

Git忽略的文件

ES语法转换

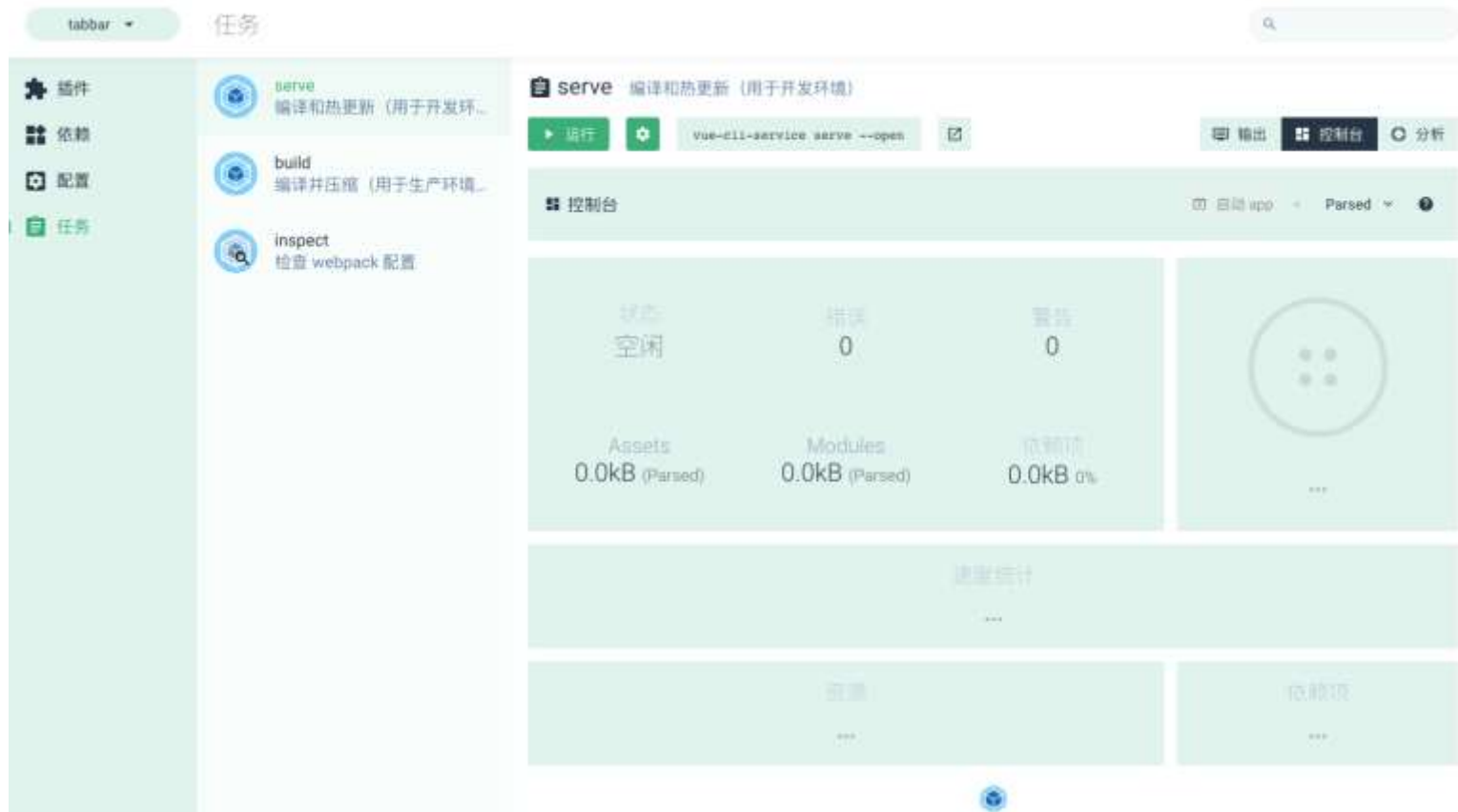
CSS相关转换

made by coderwhy
微博: coderwhy

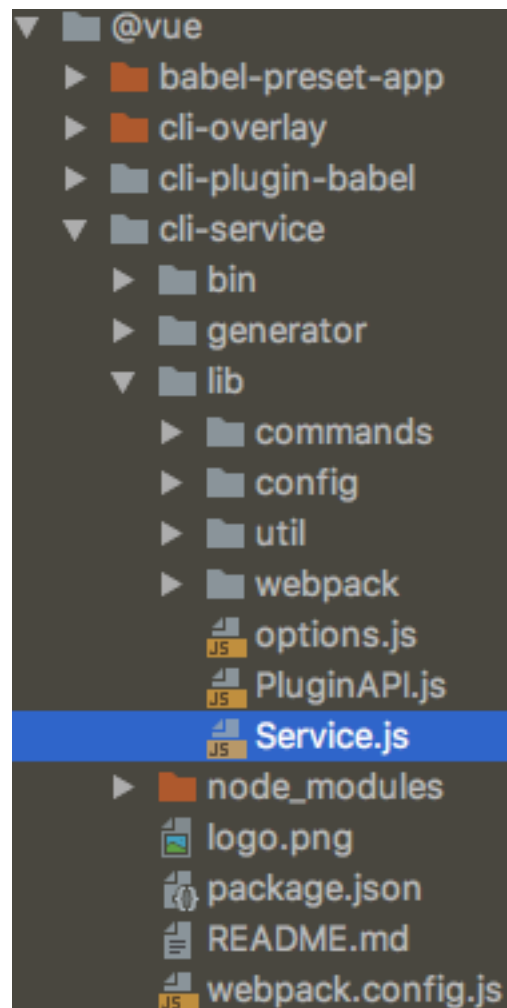
配置去哪里了?

■ UI方面的配置

□ 启动配置服务器: vue ui



■ 一大堆配置文件去哪里了?



自定义配置：起别名

```
const path = require('path')
function resolve (dir) {
  return path.join(__dirname, dir)
}

// made by coderwhy
// 微博: coderwhy

module.exports = {
  // 1.基础的配置方式
  configureWebpack: {
    resolve: {
      alias: {
        'components': '@components',
        'pages': '@pages'
      }
    }
  },
  // 2.利用webpack4的webpack-chain来配置
  chainWebpack: (config) => {
    config.resolve.alias
      .set('@$', resolve('src'))
      .set('components', resolve('src/components'))
  }
}
```