目录

[Vue-cli框架基本搭建 2](#_Toc70064132)

[Vue Router环境搭建与基本运用 6](#_Toc70064133)

[vue-axios环境搭建与基本运用 8](#_Toc70064134)

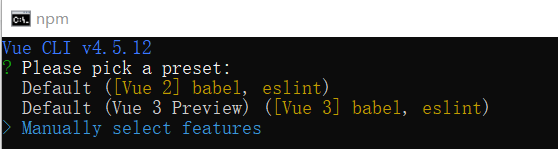
# Vue-cli框架基本搭建

#### 由创建文件路径进入cmd

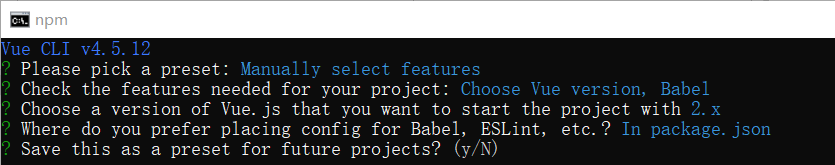
查看node是否安装：node -v

卸载指令：npm uninstall -g vue-cli  
安装指令：npm install -g @vue/cli

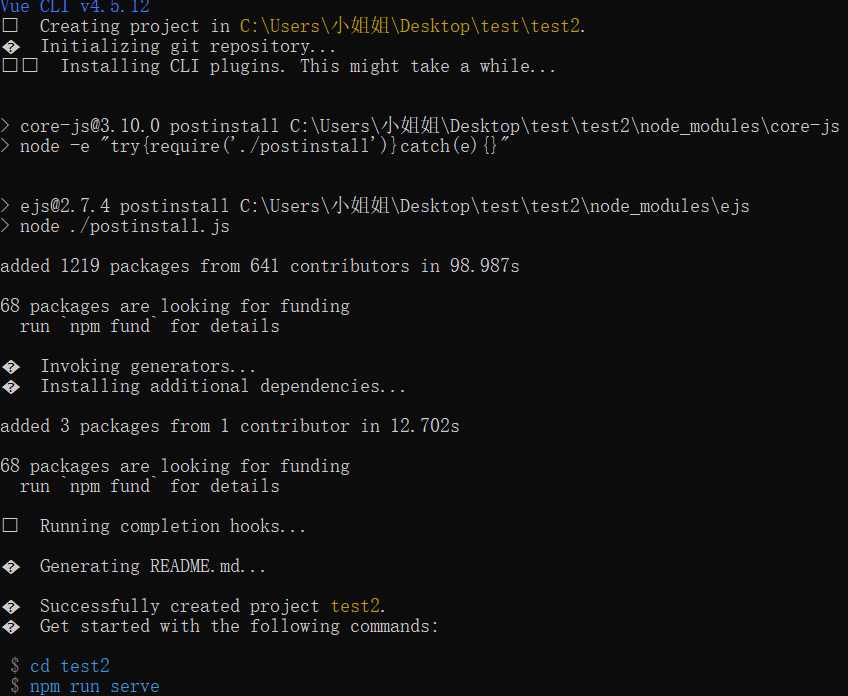
（以上步骤下次不用重复）  
创建test ：vue create test（项目名称）



注意：选第三个，第一二个少用

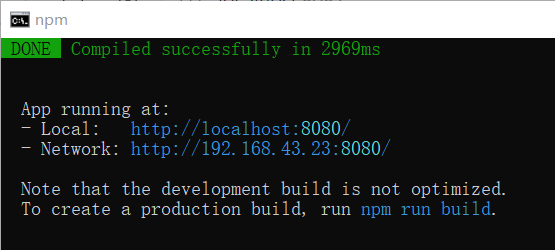


手动配置：选择（上下键，取消为空格，最后enter默认为NO）



备注：若有停顿，可以按enter或检查网络状况

若想重装请先卸载\*

进入：cd test  
npm run serve复制链接在网站上  


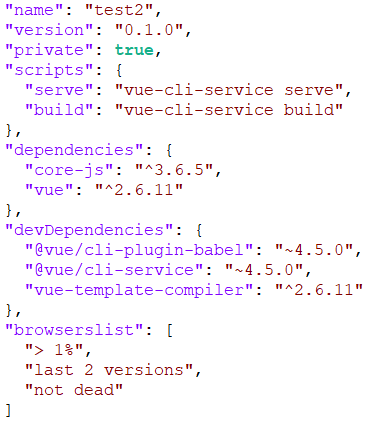


Test2是别人写的拉下来的文件，用别人的要删掉git（可能隐藏）

-node\_modules：上万个，不能拷贝，上传时要删掉

（删除后恢复npm install）

Package.json命令清单，可重新恢复上次删除的，用网络请求路径下载



.gitignore：清单

-src：源代码文件

VScode：  
集成终端npm i element-ui -S

修改Login.vue  
三部分：

模板，脚本，样式

修改APP.vue  
由上改相应（三四个名字）

<template>

  <div id="app">

    <Login/>

  </div>

</template>

<script>

import Login from './components/Login.vue'

export default {

  name: 'App',

  components: {

    Login

  }

}

</script>

##### 修改main.js

添加引用入口js（饿了么组件导入）

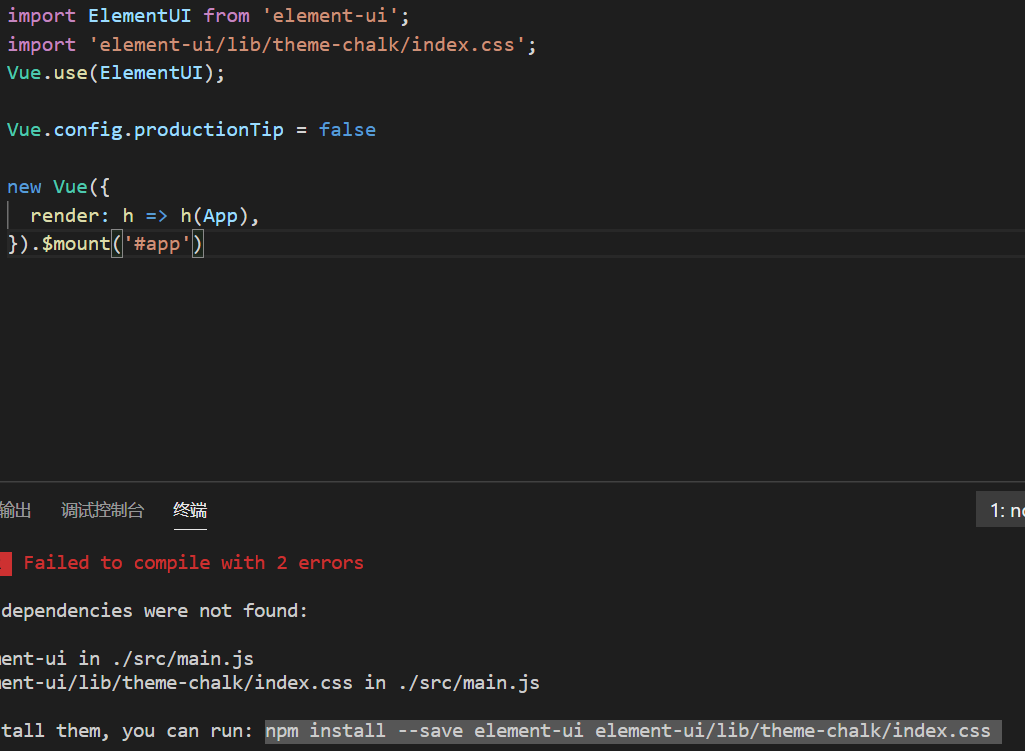
import ElementUI from 'element-ui';

import 'element-ui/lib/theme-chalk/index.css';

Vue.use(ElementUI);

<https://element.eleme.cn/#/zh-CN/component/quickstart>

集成终端：

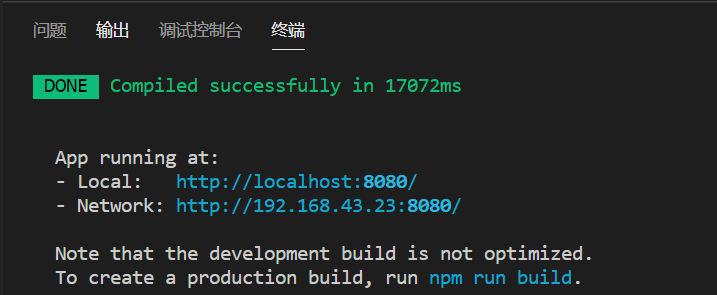


另开终端输入命令：npm install --save element-ui element-ui/lib/theme-chalk/index.css



npm run serve

注意：命令要在工程目录下（test1）运行



# [Vue Router](https://router.vuejs.org/zh/)环境搭建与基本运用

1，NPM

npm install vue-router

2，如果在一个模块化工程中使用它，必须要通过 Vue.use() 明确地安装路由功能：

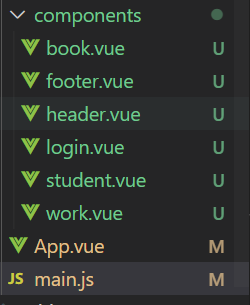
import Vue from 'vue'（注意，若第一行有则要去掉，否则报错）

import VueRouter from 'vue-router'

Vue.use(VueRouter)

<https://router.vuejs.org/zh/installation.html>

3，创建模块（login.vue work.vue ……）



4，在main.js内配置<https://router.vuejs.org/zh/guide/#javascript>

// vue-router

import VueRouter from "vue-router";

// 引入组件

import Login from './components/login.vue'

import Work from './components/work.vue'

import Student from './components/student.vue'

import Book from './components/book.vue'

// 要告诉 vue 使用 vueRouter

// 如果使用模块化机制编程，导入Vue和VueRouter，要调用 Vue.use(VueRouter)

Vue.use(VueRouter);

// 2. 定义路由

// 每个路由应该映射一个组件。 其中"component" 可以是

// 通过 Vue.extend() 创建的组件构造器，

// 或者，只是一个组件配置对象。

const routes = [

    {

        path: "/login",

        component: Login

    },

    {

        path: "/work",

        component: Work

    },

    {

        path: "/student",

        component: Student

    },

    {

        path: "/book",

        component: Book

    }

  ]

// 3. 创建 router 实例，然后传 `routes` 配置

var router = new VueRouter({

    routes

})

export default router;

// 4. 创建和挂载根实例。

// 记得要通过 router 配置参数注入路由，

// 从而让整个应用都有路由功能

new Vue({

  render: h => h(App),

  router,   //注入到根实例中

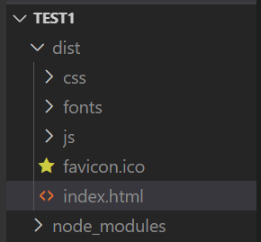
}).$mount('#app')

打包：

Vue底层仍为HTML，css，js，所以交付时：

1，终端：-npm run build （打包）

2，生成：dist文件



# vue-axios环境搭建与基本运用

<http://www.axios-js.com/zh-cn/docs/>

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中

一，安装:

CommonJS: npm install --save axios vue-axios

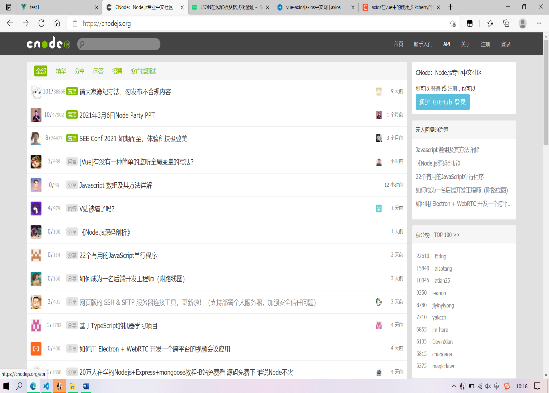
二，测试准备：

1. 找到一个暴露API网站（例：cnode）

<https://cnodejs.org/>

<https://cnodejs.org/api>

注意：以下 api 路径均以 https://cnodejs.org/api/v1 为前缀





2，转换为json格式更好看（网上在线转换格式）

<https://www.json.cn/>

三，应用：

（v-router基础上）

1，显现（header.vue）

<router-link to="/CNNode"><el-button icon="el-icon-search" round>Cnode</el-button></router-link>

2，创建CNNode.Vue

<template>

<div id="cnnode">

cnnode论坛

<li v-for='(value, index2) in this.topicsArr' :key="index2">

{{ value.title }}

</li>

</div>

</template>

<script>

import Axios from "axios";

export default {

data() {

return {

topicsArr: [],

};

},

created() {

// console.log("123");

/\* 请求数据 get返回对象\*/

Axios.get("https://cnodejs.org/api/v1/topics").then((response) => {

let data = response.data;

// let topicsArr=[];

// 用回调函数

let that = this;

if (data.success) {

that.topicsArr = data.data;

console.log(that.topicsArr);

}

});

},

};

</script>

<style>

</style>

3，结果预览：



4，自己调样式

参考资料：<https://blog.csdn.net/weixin_41777814/article/details/115178950>

mock一个简单数据类似json（上）

类似案例学习：Vue cli写的一款PC端音乐播放器（网易云的API）

<https://blog.csdn.net/L1147484597/article/details/115030977>