

1. 准备工作

Created By CGW

1.1 nodejs 下载(npm环境)

H2

注意：node版本必须大于8.9

下载地址：

```
1 | https://nodejs.org
```

1.2 安装cnpm

H2

由于NPM的仓库部署在国外,资源传输速度较慢，所以可以使用淘宝的镜像源CNPM，相当于NPM的镜像。

以下命令需要在管理员终端运行：

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

问题：

如果cnpm 出现：无法加载文件 D:\nodejs\node_global\cnpm.ps1

可以在 windows shell 使用如下命令：

```
1 | set-ExecutionPolicy RemoteSigned
```

1.3 安装调试工具vue-devtools:

H2

下载地址：<https://github.com/vuejs/devtools#/vue-devtools>

安装说明：<https://devtools.vuejs.org/guide/installation.html>

1.4 vue.js官网

H2

vue2:<https://cn.vuejs.org/v2/guide/installation.html>

其他版本:<https://cdn.jsdelivr.net/npm/vue@next/dist/> (开发版为global.js 生产版为global.prod.js)

1.5 导入vue.js库

H2

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6
7   <!-- <script src="/js/vue.min.js"></script>-->
8   <!--使用本地方式来使用Vue框架-->
9   <script src="js/vue.js"></script>
10  <!--使用CDN方式来使用Vue框架-->
11  <!-- <script src="https://unpkg.com/vue@next"></script>-->
12 </head>

```

1.6 NPM安装vue用于打包

H2

在用Vue构建大型应用时推荐使用NPM安装。NPM能很好地和诸如webpack或Browserify模块打包器配合使用。同时Vue也提供了配套工具来开发单文件组件。

由于NPM安装速度慢，推荐使用淘宝NPM镜像CNPM。

```

1 | cnpm install vue

```

1.7 命令行工具CLI的安装

H2

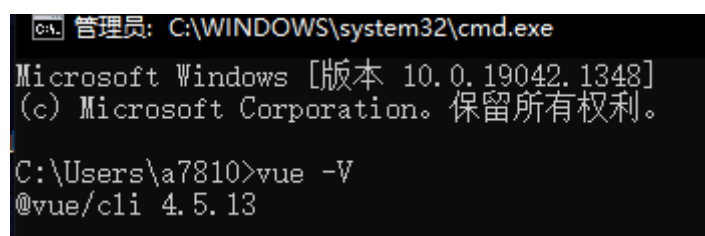
Vue-cli是用Node编写的命令行工具，需要进行全局安装，如果**不安装Vue-cli不能通过WebStorm创建Vue项目**。

```

1 | cnpm install -g vue-cli

```

安装完成如下图所示：



```

管理员: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.19042.1348]
(c) Microsoft Corporation。保留所有权利。

C:\Users\A7810>vue -V
@vue/cli 4.5.13

```

1.8 第一个Vue项目

H2

1.8.1 在单个html中使用vue

H3

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <!-- <script src="/js/vue.min.js"></script>-->
7   <script src="js/vue.js"></script>
8   <!-- <script src="https://unpkg.com/vue@next"></script>-->
9 </head>
10 <body>
11
12   <div id="app">
13     <div></div>
14     <h2>{{ explain }}</h2>
15   </div>

```

```

16
17 <script>
18   const vm = Vue.createApp({
19     data(){
20       return{
21         url:"/rsc/img/1.jpg",
22         explain:`this is a test`,
23       }
24     }
25   }).mount('#app');
26 </script>
27 </body>
28 </html>

```

1.8.2 使用命令行创建vue项目

H3

1. vue init webpack my-project
2. 进入项目文件夹my-project，然后使用：cnpm install
- 3 注意：安装依赖的时候，选择最后一个就是自己安装，用cnpm安装速度快
- 4 具体流程如下图所示

```

PS E:\Code\Web> vue init webpack vue-project
? Project name vue-project  项目名
? Project description ignore 项目描述
? Author 作者
? Vue build standalone 构建项目
? Install vue-router? Yes 路由
? Use ESLint to lint your code? No ESLint代码管理
? Set up unit tests No 单元测试
? Setup e2e tests with Nightwatch? No E2E测试
? Should we run `npm install` for you after the project has been created? (recommended) no
vue-cli · Generated "vue-project".

# Project initialization finished!
# =====
To get started:

  cd vue-project
  npm install (or if using yarn: yarn)
  npm run dev

Documentation can be found at https://vuejs-templates.github.io/webpack
PS E:\Code\Web>

```

这一步需要选第三个，因为使用cnpm安装速度快

```

PS E:\Code\Web> cd .\vue-project\
PS E:\Code\Web\vue-project> cnpm install
- [6/37] Installing import-from@^2.1.0

```

依赖安装完成后的目录如下所示：

```

1 | - - build          //项目构建 (Webpack)相关代码
2 | | --build.js      //生产环境构建代码
3 | | - - check-version.js //检查 Node、NPM等版本
4 | | - - dev-client.js  //热重载相关
5 | | - - dev-server.js  //构建本地服务器
6 | | - - utils.js       //构建工具相关
7 | | - - webpack.base.conf.js //Webpack 基础配置
8 | | - - webpack.dev.conf.js //Webpack 开发环境配置
9 | | - - webpack.prod.conf.js //Webpack 生产环境配置
10 | - - config         //项目开发环境配置
11 | | - - dev.env.js    //开发环境变量

```

```
12 | | - - index.js      //项目一些配置变量
13 | | - - prod.env.js   //生产环境变量
14 | | - - test.env.js   //测试环境变量
15 | - - node_modules    //所需要依赖资源
16 | - - src              //源码目录
17 | | - - assets        //存放资产文件
18 | | - - components    //Vue 公共组件
19 | | - - router        //存放路由JS文件，用于页面的跳转
20 | | - - App.vue       //页面入口文件
21 | | - - main.js       //程序入口文件，加载各种公共组件
22 | - - static          //静态文件，例如一些图片、JSON数据等
23 | - - .babelrc        //ES6语法编译配置
24 | - - .editorconfig   //定义代码格式
25 | - - .gitignore      //git上传需要忽略的文件格式
26 | - - README.md       //项目说明
27 | - - favicon.ico     //
28 | - - index.html      //入口页面
29 | - - package.json    //项目基本信息
```

注意：main.js、App.vue、index.js都是由2.x版本生成

main.js文件解释

```
1 /*在main.js中导入Vue对象*/
2 import Vue from 'vue'
3 /*导入App.vue组件，并且命名为App*/
4 import App from './App'
5 /*导入 router 路由*/
6 import router from './router'
7 Vue.config.productionTip = false
8 /*所有导入成功后，创建Vue对象，设置被绑定的节点为 ' # app','#app'是index.html文件中的
  个div*/
9 new Vue({
10 el: '# app',
11 /*将 router 设置到Vue对象中 * /
12   router,
13 /*声明一个组件App,App这个组件在一开始已经导入项目中了，但是无法直接使用，必须声明
  */
14 components: { App },
15 /*template中定义了页面模板，即在App组件中的内容渲染到 ' # app'这个div中*/
16 template:'<App/>'
17 } )
18
```

App.vue解释

App.Vue包含三部分组成：页面模板、页面脚本、页面样式。

```
1 <!--页面模板-->
2 <template>
3   <div id = "app">
4     <!--页面模板中定义了一张图片-->
5     <img src = "./assets/logo.png">
6     <!--router-view简单理解为路由占位符，用来挂载所有的路由组件-->
7     <router-view/>
8   </div>
9 </template>
```

```

10 <!--页面脚本：面脚本用来实现数据初始化、事件处理等-->
11 <script>
12   export default {
13     name: 'App'
14   }
15 </script>
16 <!--页面样式-->
17 <style>
18   #app {
19     font-family: 'Avenir', Helvetica, Arial, sans - serif;
20     - webkit-font-smoothing:antialiased;
21     - moz-osx-font-smoothing: grayscale;
22     text-align: center;
23     color: #2c3e50;
24     margin-top: 60px;
25   }
26 < / style>

```

index.js解释

```

1 import Vue from 'vue'
2 import Router from 'vue - router'
3 import HelloWorld from '@components/HelloWorld'
4 Vue.use(Router)
5 export default new Router({
6   /*路由文件， path路径， 对应的组件为HelloWorld,即在浏览器地址为 / 时， 在router-view
   位置显示HelloWorld组件*/
7   routes:[
8     {
9       path: '/',
10      name: 'HelloWorld',
11      component:HelloWorld
12    }
13  ]
14 })

```

1.8.2.1 项目运行

H4

```

1 | cnpn run dev

```



Welcome to Your Vue.js App

Essential Links

[Core Docs](#) [Forum](#) [Community Chat](#) [Twitter](#)
[Docs for This Template](#)

Ecosystem

[vue-router](#) [vuex](#) [vue-loader](#) [awesome-vue](#)

1.8.2.2 项目打包

H4

```
1 | cnpm run build
```

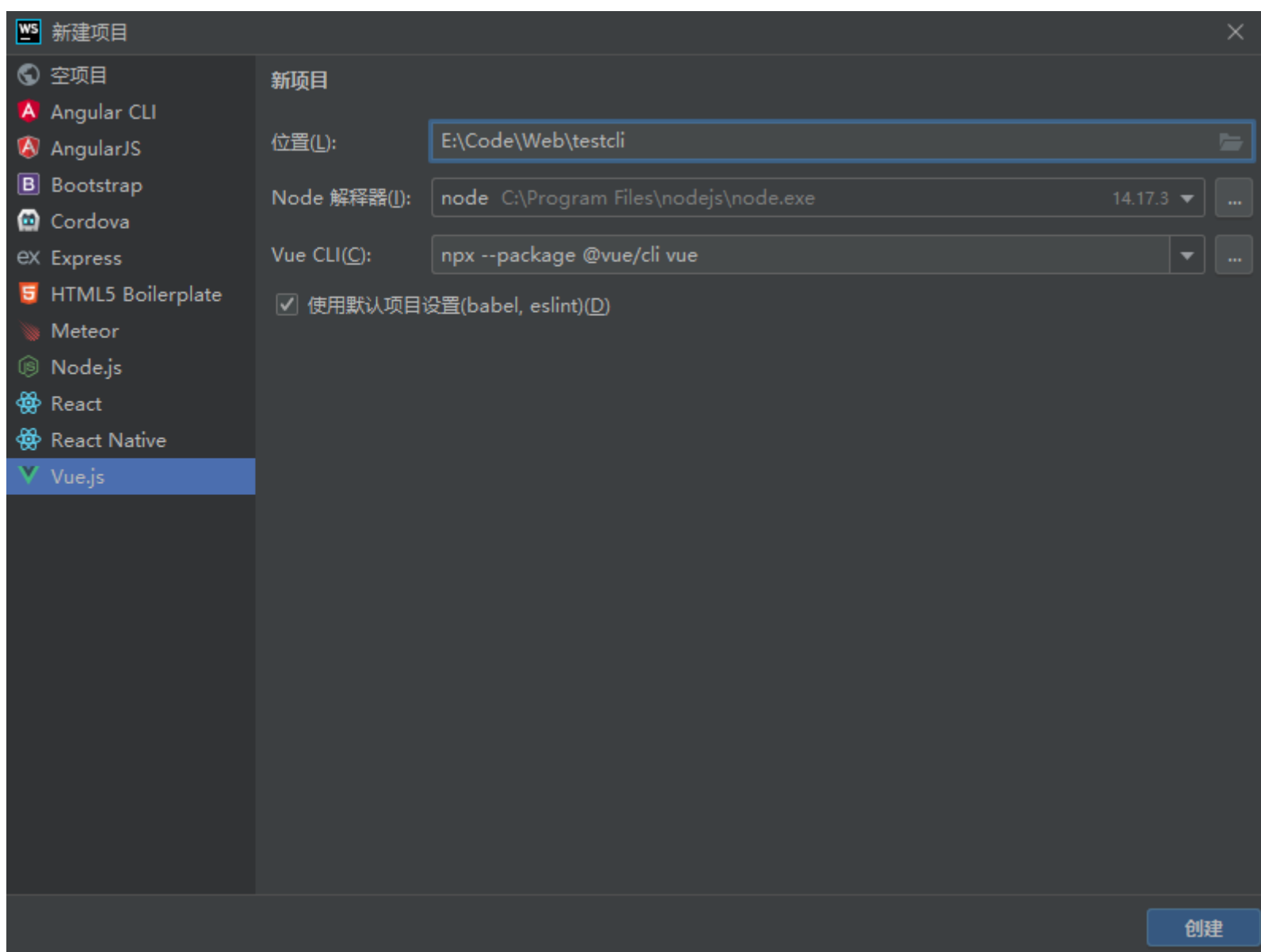
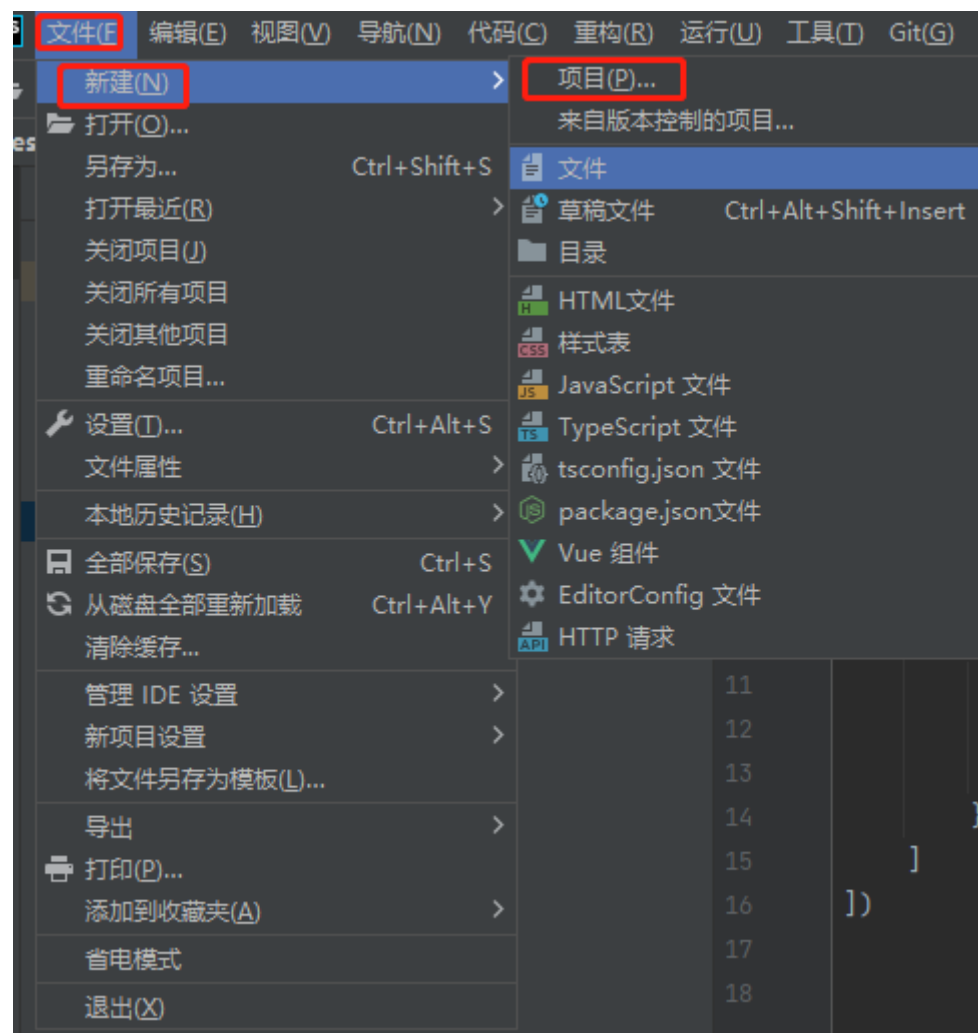
打包完会生成一个dist的打包文件：

看

资料盘 (E:) > Code > Web > vue-project					build	▼	🔄	搜索"build"
名称	修改日期	类型	大小					
WS build.js	2021/12/17 21:12	JetBrains WebSt...	2 KB					
WS check-versions.js	2021/12/17 21:12	JetBrains WebSt...	2 KB					
logo.png	2021/12/17 21:12	PNG 文件	7 KB					
WS utils.js	2021/12/17 21:12	JetBrains WebSt...	3 KB					
WS vue-loader.conf.js	2021/12/17 21:12	JetBrains WebSt...	1 KB					
WS webpack.base.conf.js	2021/12/17 21:12	JetBrains WebSt...	2 KB					
WS webpack.dev.conf.js	2021/12/17 21:12	JetBrains WebSt...	3 KB					
WS webpack.prod.conf.js	2021/12/17 21:12	JetBrains WebSt...	5 KB					

1.8.3 使用Webstorm创建vue项目

H3



如果出现 **npm ERR! or that you lack permissions to access it.**问题，删除C:\Users{账户}\下的.npmrc文件，新建项目即可。

1.8.4 命令行创建，然后使用web storm打开项目

H3

注意：使用Webstorm创建vue项目可能会比较慢，所以可以用命令行创建项目然后再用Webstorm打开，步骤如下图所示：

