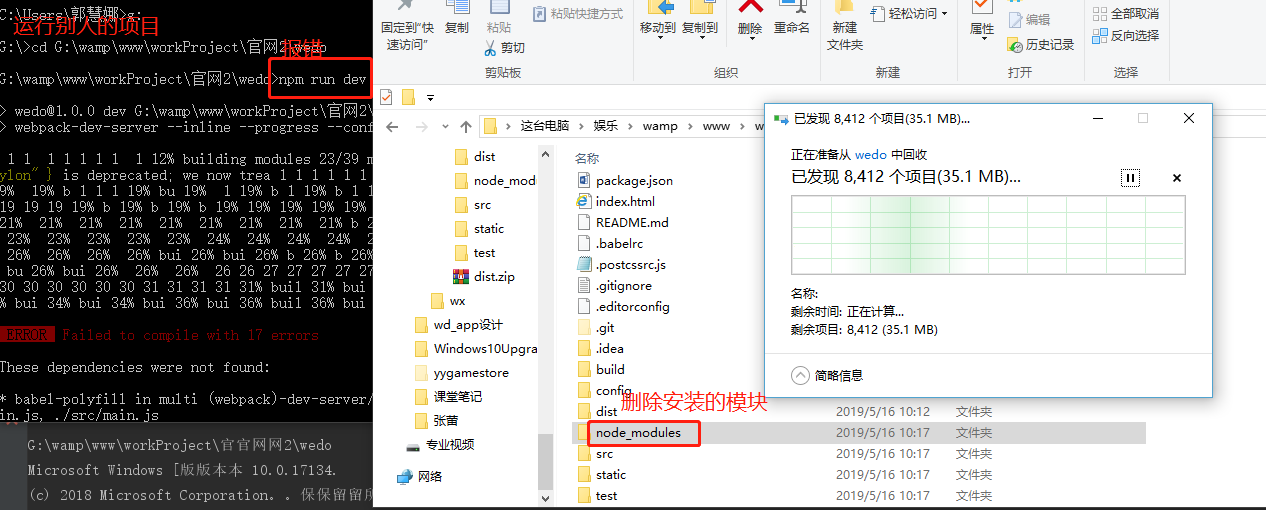
# VUE-CLI

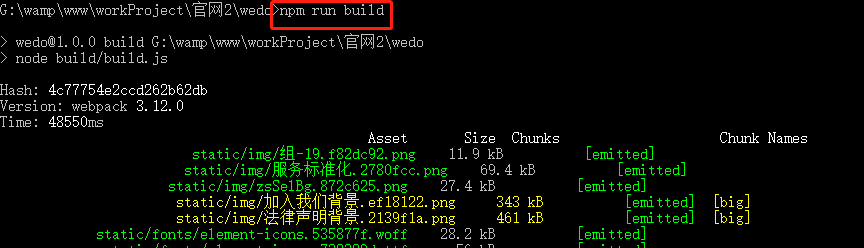
## 一交接项目

1将别人项目复制过来。

2重新安装npm install



3重新打包



4在项目目录中，运行命令 npm run dev

## 二脚手架

### Vue安装

一安装VUE

安装nodejs 6.9以上 ，npm3.10以上

1. 需要使用npm，全局安装webpack，由于从webpack@4.0 开始需要安装webpack-cli

先CMD执行，webpack -v确认是否全局安装过

没有的话执行：

1）npm install -g webpack / npm install webpack -g

在全局安装 webpack-cli 依赖

2） npm install webpack-cli -g

3） 全局安装 vue-cli：npm i -g vue-cli

或者：npm install --globel vue-cli

安装完成后确认是否安装CMD： vue-V

4）CMD：vue-list: 列出可用的模板

2. 利用vue-cli 脚手架构建项目

1） 首先建立一个 叫 demo的文件夹

2）然后：cd demo 将构建的项目放进demo 文件夹

3） 执行： vue init webpack 项目名称

4）执行后显示

? Project name 项目名称

?Project description 项目描述 Y

Author 作者 Y

推荐的运行：Runtime-only: about 6KB lighter min+gzip, but templates (or any Vue-specific HTML) are ONLY allowed in .vue files - re

nder functions are required elsewhere： 推荐的运行加编译

Install vue-router：安装路由

Use ESLint to lint your code? 是否使用ESLint管理代码（一般选择no）

Set up unit tests：是否建立单元测试

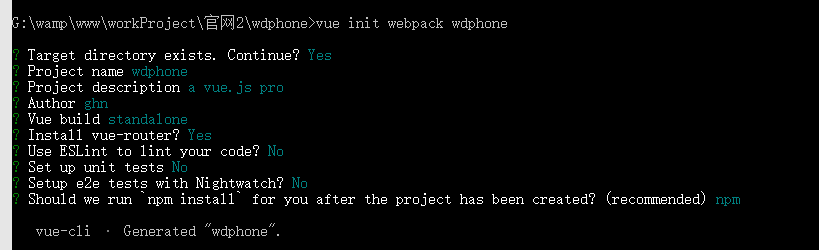
Setup e2e tests with Nightwatch?: 是否安装对端测试

vue init webpack 项目名称：适用于中大型项目

vue init webpack-simple 项目名称: 小型项目

5）然后：cd 项目名称

执行：npm run dev 运行项目占用8080



3 打包上线：

自己的项目放在，src文件夹下

编辑开发完成后打包在CMD中执行：npm run build

打包完成后会默认生成 dist 文件夹

项目上线，只需要将dist 文件夹交给后台人员放在服务器上就好了

二VueCLI脚手架

脚手架是通过webpack搭建的开放环境

使用ES6语法

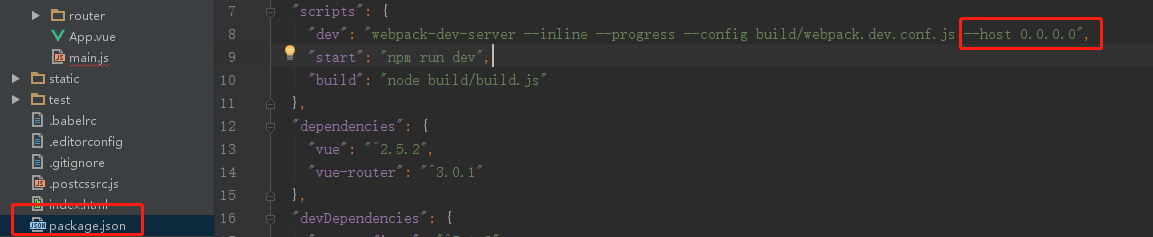
打包和压缩JS为一个文件

项目文件在环境中编译，而不是浏览器

实现页面自动刷新

三配置局域网IP

package.json 文件中添加 –host 0.0.0.0



npm run dev

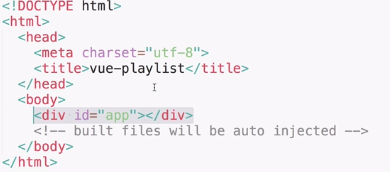


### 根组件

src -assets 可以放图片。

components 组件。

index.html 当前文件的入口文件。可以通过动态的js插入到此容器中。



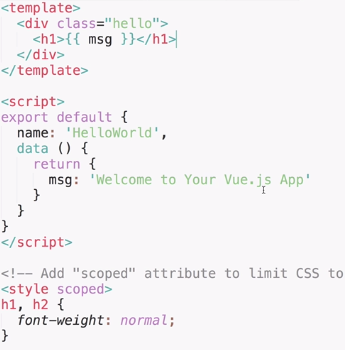
main.js ---当执行完index.html文件就会执行main.js文件。



App.vue 组件（模板，行为，样式）



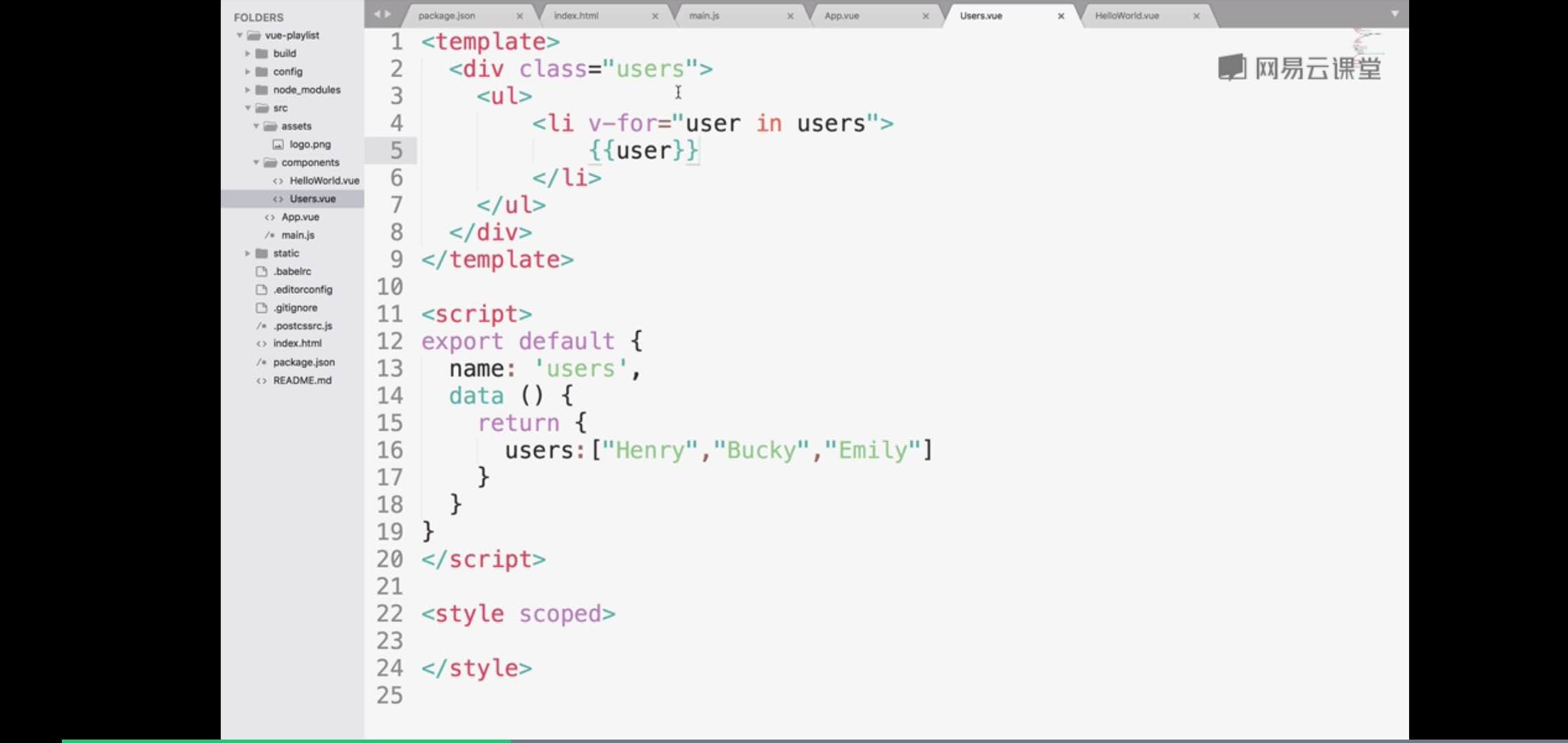
Helloworld.vue



### 全局注册组件



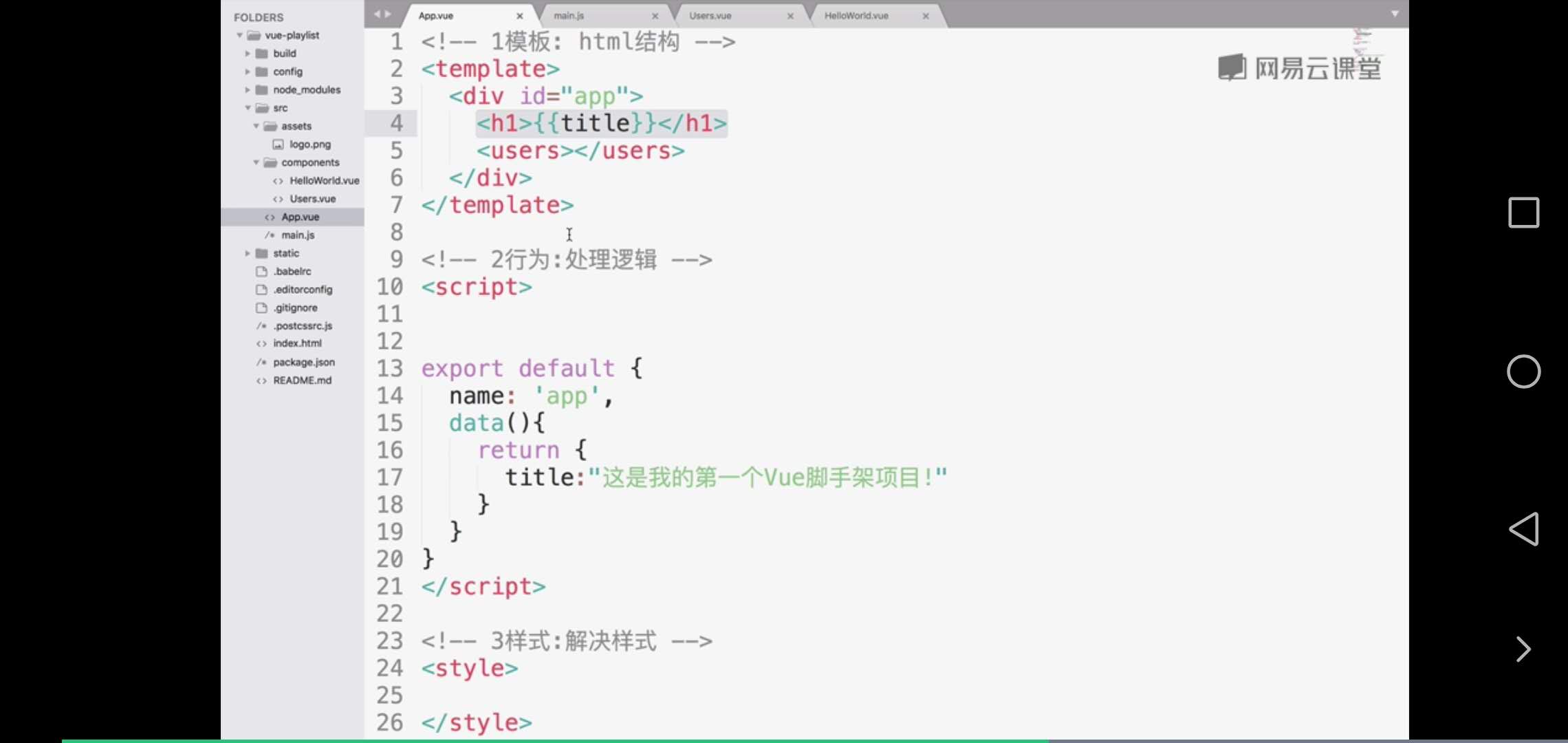
* 1. 在components文件夹下面新建Users.vue文件用来写组件。其中【name】是你的组件名字。



* 1. 在main.js中导入组件的名字。使用Vue.component(“users”,”Users”)全局注册件。



* 1. 在App.vue中使用组件



### 局部注册组件

* 1. 在App.vue中导入Users文件，然后注册组件即可。Components:{“users”:Users}

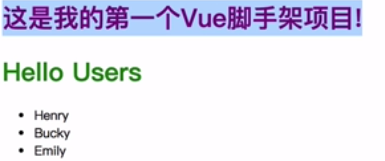


* 1. 在main.js中先注释掉注册的全局users组件。



### css样式

style中添加Scoped 域来实现子样式



### 组件嵌套

* 1. 在components文件下新建footer/header组件和usert.vue组件

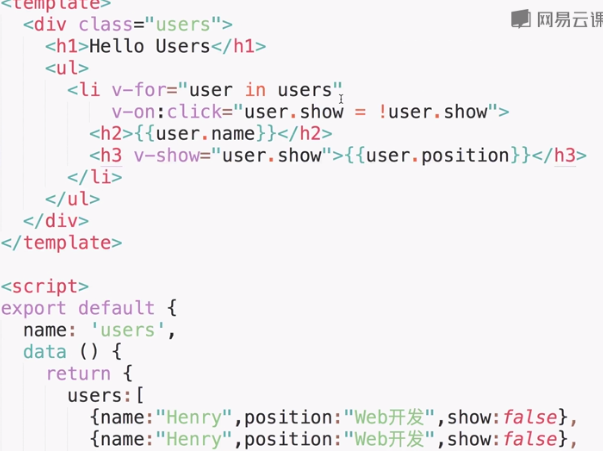
Footer.vue



Header.vue



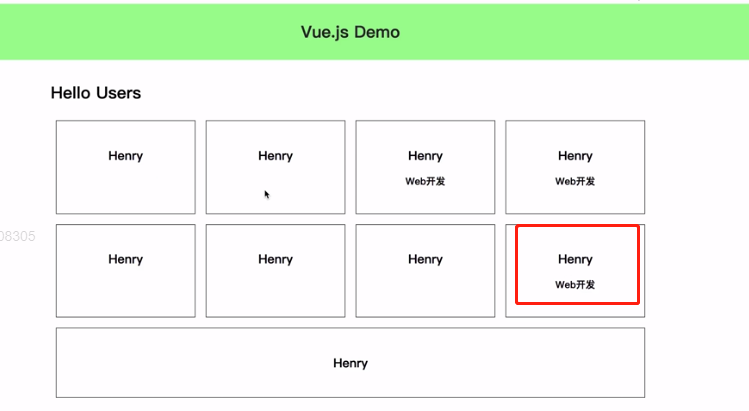
User.vue



* 1. 在App.vue中注册局部组件。并引用。





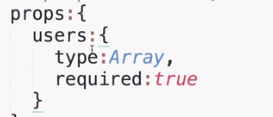


### 父-子【属性传值Props】

1. 父组件向子组件传值。将User.vue中的data剪切到App.vue的data中。使用v-bind绑定div的属性。

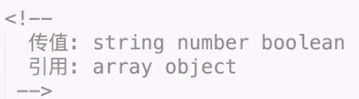


1. 在User.vue中，使用props:[“div属性名”]的方式获得数据或则使用官方提供的标准方式。如下图。





### 父-子Vue【传值和传引用】



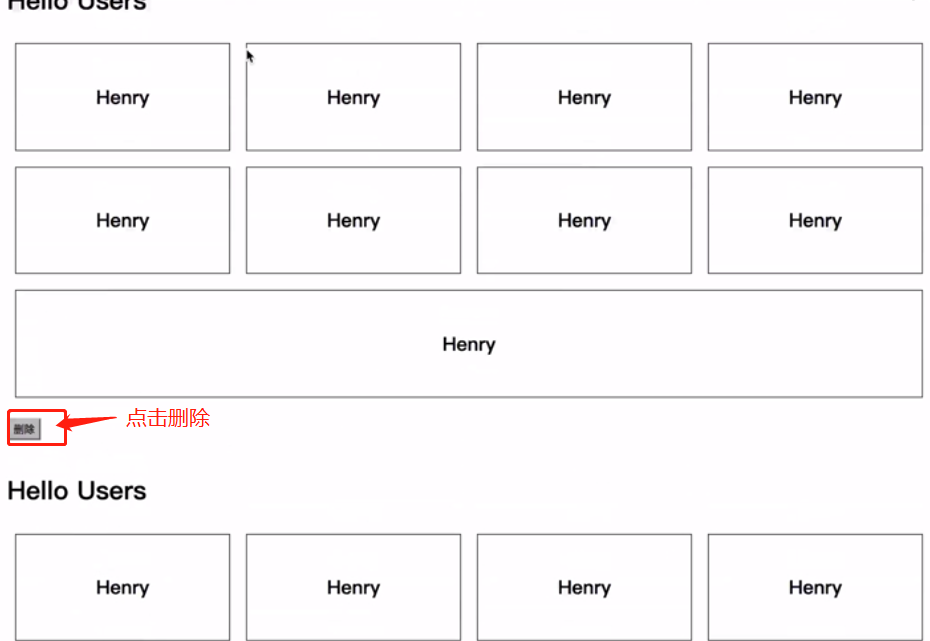
* 1. 传引用。
     1. APP.vue中，使用v-bind绑定div属性

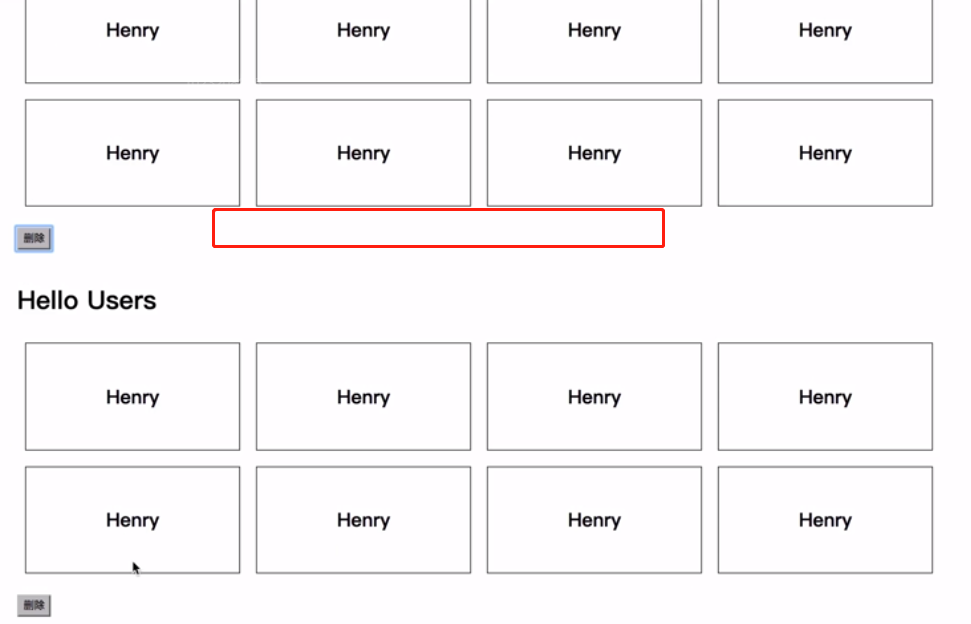




2）在User.vue中，接受传递的值，并且添加删除按钮





结果：发现两个值【herry】都被删除了。

② 在Header.vue/Foot中接受值传递。给header添加点击事件之后，发现点击头部之后，头部的值改变，底部的值没有改变。







重点：传值一般都是单个值类型，比如String，Number，boolean

传引用一般都是Array，Object

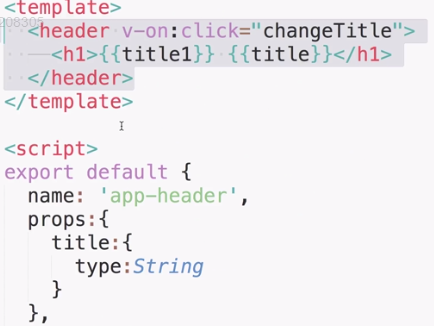
### 子-父【事件传值】

效果：点击绿色区域，发现Header内容改变为第二张图。并且Footer的内容也改变了。



方法：在Header.vue中。点击changTitle的方法，触发方法【方法中注册了titleChange事件，并携带了参数】，它将回去父级去找titleChanged事件，发现需要执行updateTitle方法【$event用来接受参数】，将会改变title的值并赋值给title，所以v-bind:title绑定的方法将会改变。Header.vue和Footer.vue将会变化。







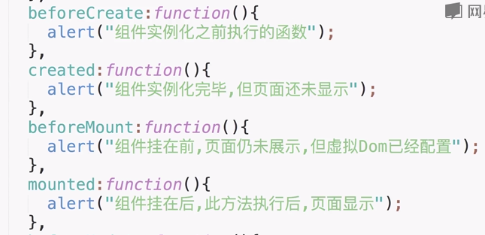


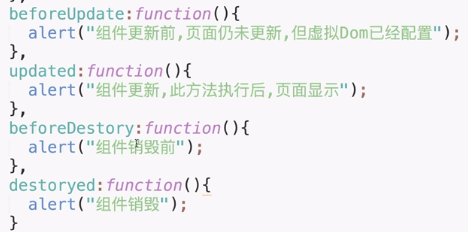
### 生命周期



钩子函数8个

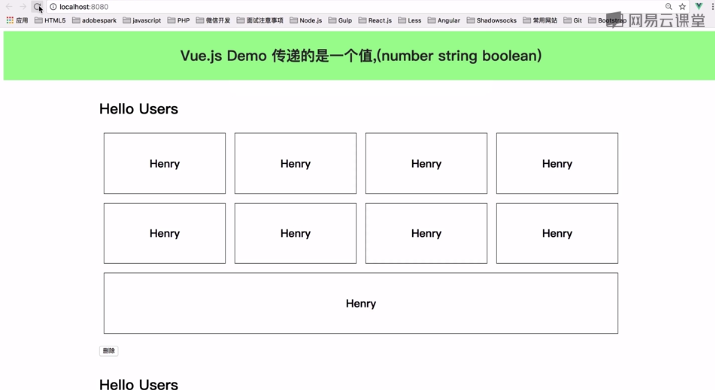
1.

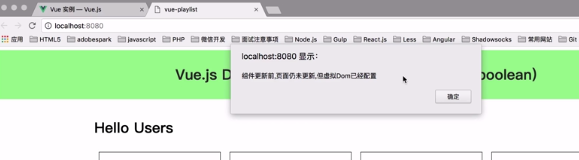




效果Mounded挂在后，页面显示。







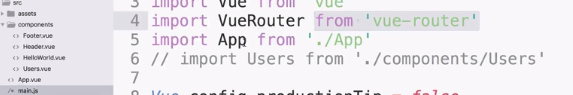
### 11. 路由和Http

**一、路由好处**，不用刷新，直接跳转到所需路径。使用a标签的化跳转需要请求网络。

1. npm install vue-router –-save-dev

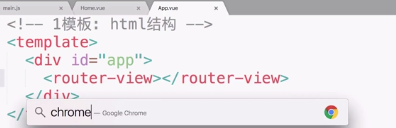
2. npm run dev

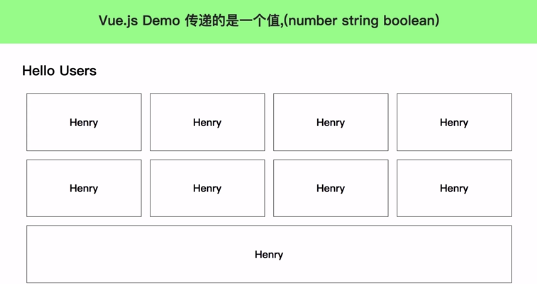
3. main.js引入路由/使用路由Vue.use(VueRouter)/配置路由new VueRouter/注册路由





4在app.vue中使用路由。



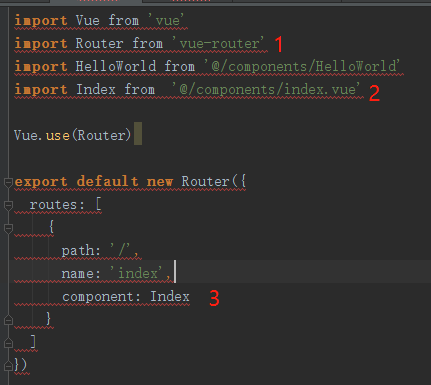


Eg:点击home和hello分别跳转到相对目录。

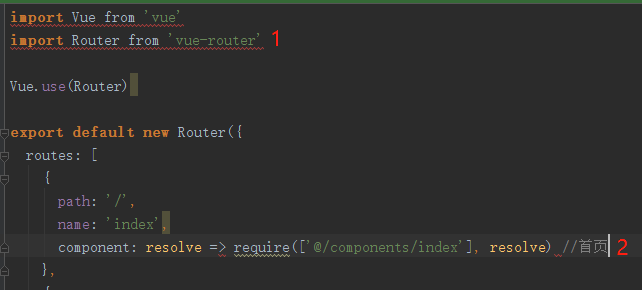


**二、使用路由**

1. import路由



1. 使用resolve

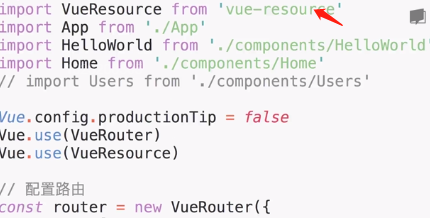


二、http

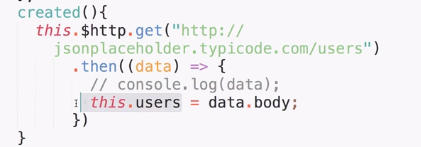
1. 安装 npm install vue-resouce –-save-dev

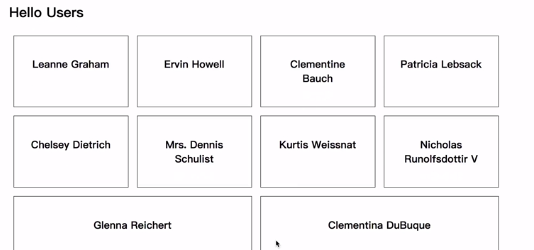
2．重启npm run dev.

3. 在main.js中引入resouce/使用



4.在Home.vue 组件中使用created（页面加载之前）函数中请求数据并赋值给页面。





### 11 .vue切换router跳转之间，页面滚动位置相互影响。

1. 在 app.vue中监听路由

//app.vue

export default {

watch:{

'$route':function(to,from){

　　　　　　　　　　 document.body.scrollTop = 0；

document.documentElement.scrollTop = 0;

}

}

}

1. 全局路由卫士

// router.js

router.afterEach(() => {

document.body.scrollTop = 0;

document.documentElement.scrollTop = 0;

})

### 12.安装jquery插件

1． cmd下载

npm install jquery –save

2． webpack.base.conf.js添加如下内容

**1.引入'webpack'**

**const** webpack = require('webpack') //引入jq

**2.在module.exports 处导入插件**

plugins:[  
 **new** webpack.ProvidePlugin({  
 $:'jquery',  
 jQuery:'jquery',  
 "window.JQuery":"jquery"  
 })  
]

3 .使用

### 13.安装less插件

1cmd 下载

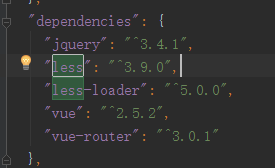
npm i less less-loader –S

2 在webpack.base.conf.js添加如下内容

**module.exports中添加**

{  
 test: /\.less$/,  
 loader: "style-loader!css-loader!less-loader"  
}

3重新运行之后。在package.json文件中发现如下



### 14 .安装swiper插件

* + 1. cmd安装，之后npm run dev即可npm install swiper --save-dev
    2. 使用swiper-"version": "4.5.0"发现轮播图不能出现分页器，并且自动轮播失效。

CSS中添加

<!-- 如果需要分页器 -->

<div class="swiper-pagination"></div>

页面加载前挂载

import "swiper/dist/css/swiper.css";

import Swiper from "swiper";

export default {

mounted() {

this.$nextTick(() => {

new Swiper(".swiper-container", {

loop: true, // 循环模式选项

autoplay: true, // 自动滑动

pagination: {

el: ".swiper-pagination"

}

});

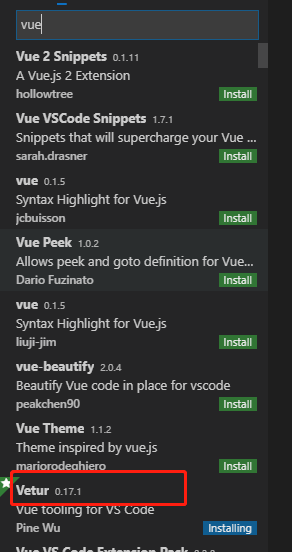
});

}

}

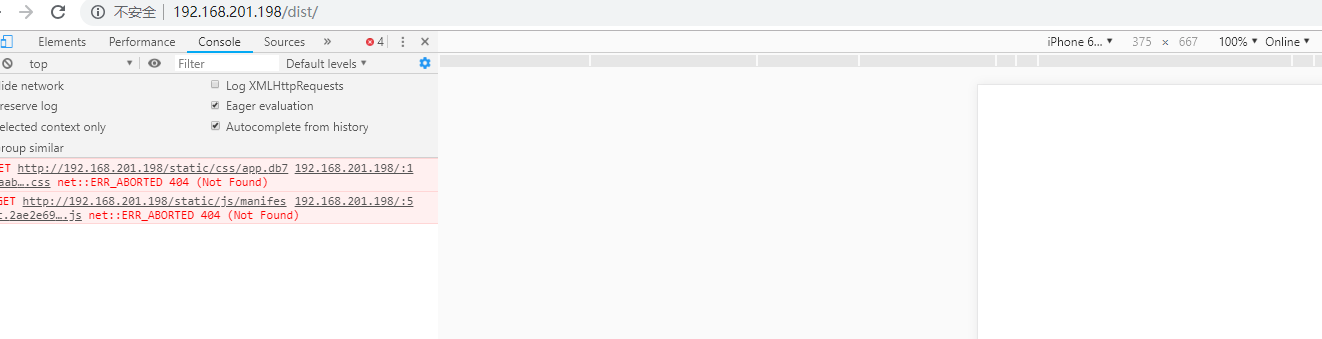
### 15.vscode编辑器

扩展vue文件，安装vetur插件

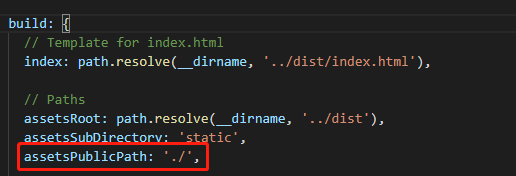


## 三打包

1.Npm run build 之后出现空白页



1. 在config中添加./



2）重新打包即可

2.background-image图片路径404。

