# Vue 使用教程

学习资料：

1. VueJS: <https://cn.vuejs.org/v2/guide/>
2. Vue-Router: <https://router.vuejs.org/zh/>
3. VUEX: <https://segmentfault.com/a/1190000009404727>
4. ES6: <http://es6.ruanyifeng.com/#docs/style>
5. SCSS: <https://www.sass.hk/docs/>
6. Element(UI): <https://element.eleme.io/#/>
7. iView(UI): <https://www.iviewui.com/>
8. axios(网络请求): <https://www.kancloud.cn/yunye/axios/234845>

目录

[Vue 使用教程 1](#_Toc536461478)

[一、 环境搭建 3](#_Toc536461479)

[二、 创建项目 5](#_Toc536461480)

[三、 使用WebStorm运行 9](#_Toc536461493)

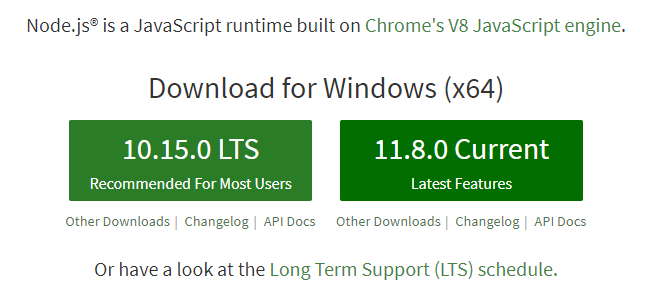
[四、 .vue 文件结构 12](#_Toc536461496)

[五、 Hello World 13](#_Toc536461497)

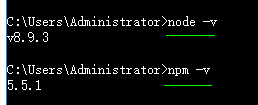
[六、 引入第三方库 14](#_Toc536461498)

## 环境搭建

下载NodeJS： <https://nodejs.org/en/> 选择最新稳定版本下载并安装。



打开CMD，检查Node环境是否正常：



NPM是基于NodeJS的一个仓库，使用NPM可以省去很多以前在Jquery项目中找插件、更新插件版本、管理插件等繁杂的工作。

如一切正常，就可以使用命令：

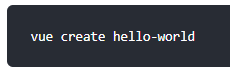
安装Vue（window中去掉$）

安装Vue-Cli：（Cli 一般指“脚手架”工具，是使用命令行搭建项目基本框架的工具）



## 创建项目

安装好Vue-Cli之后，直接在控制台输入 vue create 项目名，以创建一个新项目。



以下参考：<https://gitee.com/hjm100/codes/rjch7b31l4f59gt8suidn63>

### 选择项目类型

**Vue** **CLI** v3.**0.0**-beta.6

? **Please** pick a preset: (**Use** arrow keys)

> xs-**default** (**vue**-**router**, **vuex**, **stylus**, **babel**, **pwa**, **eslint**, **unit**-**jest**) // 这是我运行过之后的默认设置，第一次执行create是没有的

**default** (**babel**, **eslint**)

**Manually** select features

// 注：按键盘上下键选择默认（**default**）还是手动（**Manually**），

//如果选择**default**，一路回车执行下去就行了（注：现在vue-cli3.0默认使用yarn下载），这里我选择手动，

### 选择特性支持

? Please pick a preset: Manually select features

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

>( ) Babel

>( ) TypeScript  *// 支持使用 TypeScript 书写源码*

( ) Progressive Web App (PWA) Support  *// PWA 支持*

( ) Router  *// 支持 vue-router*

( ) Vuex  *// 支持 vuex*

( ) CSS Pre-processors  *// 支持 CSS 预处理器。*

( ) Linter / Formatter  *// 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。*

*// 注意：你要集成什么就选就行了（注：空格键是选中与取消，A键是全选）*

* Babel：必选，一个javascript转译器，将最新版的js语法（es6、es7）转换为现阶段浏览器可以兼容的js代码。
* typescript：作用有些类似于babel，拥有类型检查能力和面向对象新特征。
* PWA：渐进式WEB应用
* Router：Vue路由组件，基本是必选，除非使用了其他自带路由的框架。
* Vuex：Vue状态管理工具，推荐选择。
* CSS Pre-processors：CSS预处理工具，推荐使用SASS/SCSS写样式。
* Linter：语法检测工具，推荐。
* Unit Testing：单元测试
* E2E Testing：e2e测试

### 选择css预处理，这里如果需要用到的话选择SCSS/SASS

? Please pick a preset: Manually **select** features

? **Check** the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer **and** CSS Modules **are** supported **by** **default**):

> SCSS/SASS

LESS

Stylus

### 选择ESLint + Prettier

? Please pick a preset: Manually **select** features

? **Check** the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer **and** CSS Modules **are** supported **by** **default**): Stylus

? Pick a linter / formatter config: (**Use** arrow keys)

> ESLint **with** error prevention **only**

ESLint + Airbnb config

ESLint + Standard config

ESLint + Prettier

### 选择语法检查方式，这里我选择保存就检测

Vue CLI v3.**0.0**-beta.6

? Please pick a preset: Manually **select** features

? **Check** the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer **and** CSS Modules **are** supported **by** **default**): Stylus

? 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 // 保存就检测

( ) Lint **and** fix **on** **commit** // fix和**commit**时候检查

### 选择单元测试

Vue CLI v3.**0.0**-beta.6

? Please pick a preset: Manually select features

? Check the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported **by** **default**): Stylus

? Pick a linter / formatter config: Prettier

? Pick additional lint features: Lint **on** **save**

? Pick a unit testing solution: (Use arrow keys)

> Mocha + Chai

Jest

### 她会问你 ，把babel,postcss,eslint这些配置文件放哪，这里随便选，我选择放在独立文件夹

Vue CLI v3.**0.0**-beta.6

? Please pick a preset: Manually select features

? Check the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported **by** **default**): Stylus

? Pick a linter / formatter config: Prettier

? Pick additional lint features: Lint **on** **save**

? Pick a unit testing solution: Jest

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

> In dedicated config files *// 独立文件放置*

In **package**.json *// 放package.json里*

### 键入N不记录，如果键入Y需要输入保存名字，如第一步所看到的我保存的名字为xs-default

Vue CLI v3.**0.0**-beta.6

? Please pick a preset: Manually select features

? Check the features needed **for** your project: Router, Vuex, CSS Pre-processors, Linter, Unit

? Pick a CSS pre-processor (PostCSS, Autoprefixer and CSS Modules are supported **by** **default**): Stylus

? Pick a linter / formatter config: Prettier

? Pick additional lint features: Lint **on** **save**

? Pick a unit testing solution: Jest

? Where **do** you prefer placing config **for** Babel, PostCSS, ESLint, etc.? In dedicated config files

? Save **this** **as** a preset **for** future projects? (Y/n) *// 是否记录一下以便下次继续使用这套配置*

### 确定后，等待下载依赖模块

## 启动项目（使用CMD）

1. 初始完之后，进入到项目根目录： cd my-project
2. 启动项目：npm run serve
3. 在浏览器输入[http://localhost:8080就可以看到vue的欢迎界面](http://localhost:8080%E5%B0%B1%E5%8F%AF%E4%BB%A5%E7%9C%8B%E5%88%B0vue%E7%9A%84%E6%AC%A2%E8%BF%8E%E7%95%8C%E9%9D%A2/)

## 打包上线

1. vue-cli 也提供了打包的命令，在项目根目录下执行： npm run build
2. 执行完之后，可以看到在项目根目录下多出了一个 dist 目录，该目录下就是打包好的所有静态资源，直接部署到静态资源服务器就好了。

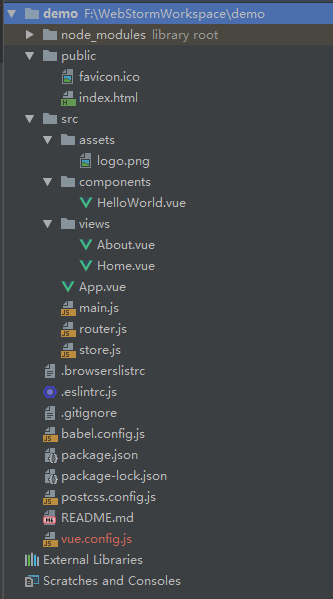
## 注意：

1. 如果想要修改端口号以及webpack配置，需要你在根目录下面创建一个vue.config.js

## 使用WebStorm运行

使用WebStorm打开刚才生成的项目文件夹。

项目的大致结构：



### 文件夹说明

* demo 项目名
  + node\_modules npm库文件，不需要管理，也不要提交至svn/git
  + dist 项目打包后的文件目录，用于部署，不需要提交至svn/git
  + public
    - index.html 项目的index页面，一般不需要改，所有内容均由打包后的app.vue加载进html中。
    - 其他公用资源文件
  + src 资源文件夹，主要的项目文件地址
    - assets 静态资源文件夹，主要存放图片等资源
    - components 自定义组件文件夹
    - views 页面文件夹
    - 以及其他类似js, css等文件夹
    - App.vue 项目根页面，一般不需要改
    - Main.js 项目根入口，主要实现一般全局初始化、模块引入等操作
    - Router.js 路由设置js
    - Store.js vuex全局管理js
  + Package.json 用于管理npm第三方库
  + Vue.config.js vue的特定配置，详见：

Vue.config.js vue的特定配置，详见：<https://gitee.com/hjm100/codes/rjch7b31l4f59gt8suidn63>

### 运行项目

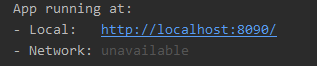
有两种方式可以运行项目：

1. 点击WebStorm左下角 Terminal（控制台），输入 npm run serve，即可运行debug模式。
2. 点击右上角的空白下拉框，选择edit configurations，在对话框中点击左上角的 “+”号，选择npm，之后在右边依次选择 run, serve, 点击 ok。右上角的图标会发生变化，两个按钮分别代表run和debug

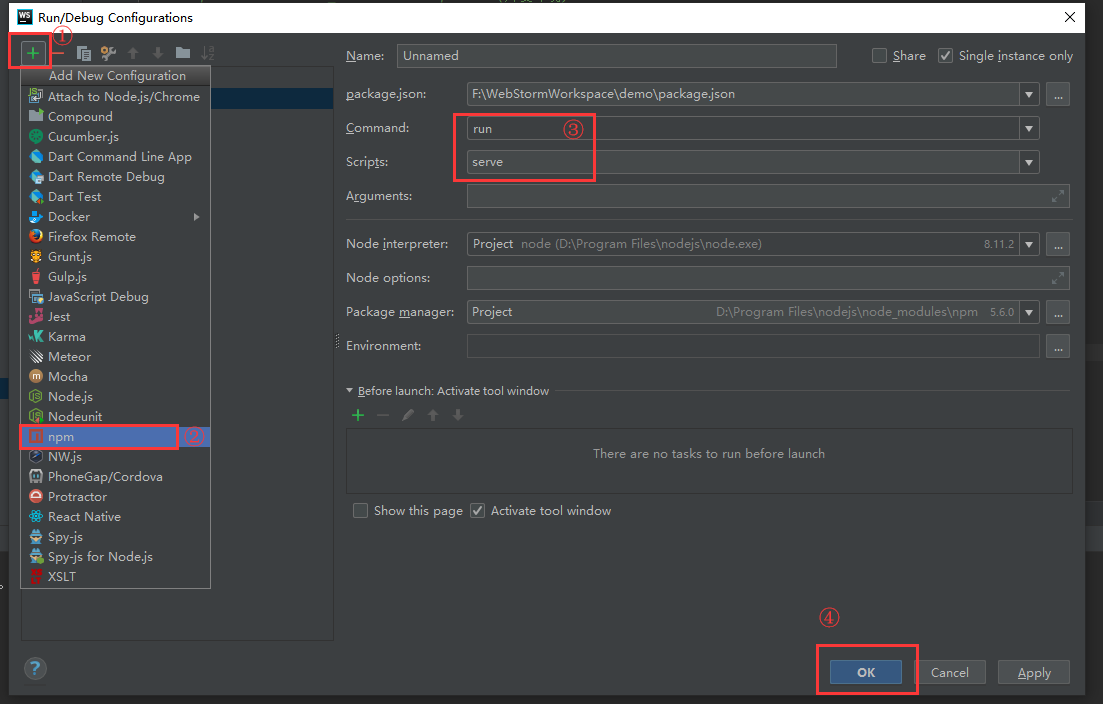
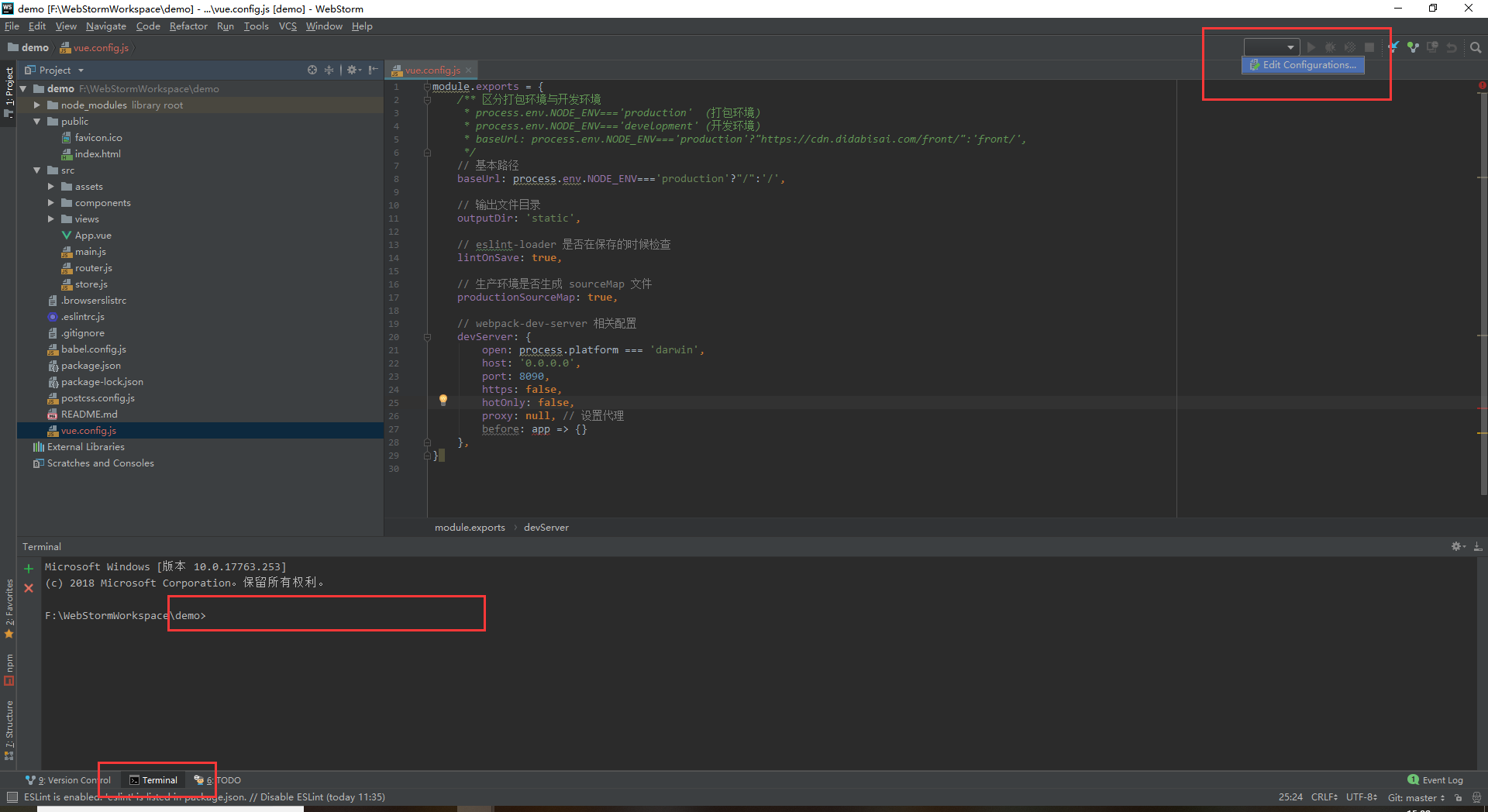


等待npm启动完毕，项目访问路径通常为 <http://localhost:8080>。

其中端口号需要在vue.config.js中进行修改。在下方run的窗口中可以找到项目访问路径，点击即可访问。



在terminal界面按下 ctrl + c （复制）将会提示是否中断运行，类似于tomcat的开启和关闭。



## .vue 文件结构

一般的 .vue 文件分为三块，如图：



被 <template>标签包裹的是HTML页面模板，基本使用方法与HTML5无异，但是有许多模板语法的特性。

被 <script>标签包裹的是页面逻辑代码，在export default {} 中主要遵循 vue的规范，详见下例，每一个字段名，官网会有更详细的解释。

被 <style>标签包裹的是样式代码，如果后面加上了 scoped参数，则表示该段css样式仅在当前文件的作用域中生效，所以一般推荐加上scoped参数。如果需要样式复用的情况，可以考虑写在公用的css/scss等文件中，或者封装成组件。

## Hello World

接下来就可以按照官网的例子，一步一步跟着做，了解并熟悉Vue的特性：

<https://cn.vuejs.org/v2/guide/index.html#%E5%A3%B0%E6%98%8E%E5%BC%8F%E6%B8%B2%E6%9F%93>

主要需要了解的点有：

* **模板语法**
* **计算属性和侦听器**
* **Class与Style绑定**
* **条件渲染**
* **列表渲染**
* **事件处理**
* **表单输入绑定**
* 组件基础
* Prop
* 插槽
* **生产环境部署**
* **路由（Vue-router）**
* **状态管理（VUEX）**

## 引入第三方库

Vue的开发过程中，第三方库是必不可少的。引入方式分为三种：

1. **下载JS文件至public 文件夹下，然后在 index.html中直接用<script>标签引入。**
2. **直接在index.html中引入网络地址。**

这两种方式引入的JS可以直接在代码中调用全局变量来使用，但是在ESLint检测下会报错，所以最好做一下配置上的处理。

新建一个 webpack.config.js, 里面写入：

module.exports = {

externals: {

jquery: 'jQuery'

//将需要忽略打包的都写在这个里面，但前提是index.html文件里面必须script引入

}

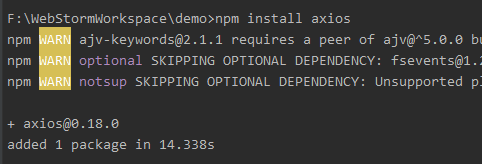
};

然后在文件中import：

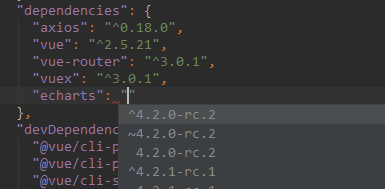
import $ from 'jquery';

1. **使用npm安装。**

有两种方式：直接在Terminal中输入： npm install 插件名



或者在package.json中手动输入插件名、版本号（会有提示），然后Webstorm会提示是否执行 npm install？ 此时可以选“是”，或者稍后手动输入 npm install 指令。



1. **自己封装的插件。**

以FileUtils.js 为例，首先需要将js文件放在src/assets/中的任意文件夹下，并且在js文件最后加入：

**export default** {  
 FileUtils, //模块名  
 install(Vue){  
 //Vue.prototype, 模块调用时的别名, 模块对象  
 Object.defineProperty(Vue.prototype, '$FileUtils', { value: FileUtils });  
 }  
};

以上两种方式引入的，需要在vue项目中调用。最简单的方式是在main.js中引入并添加：

/\*file utils\*/  
**import** FileUtils **from** './assets/js/FileUtils.js'  
Vue.use(FileUtils, '$FileUtils');

部分第三方插件需要使用其他的方式，如axios, echarts：

*/\*\* axios\*/***import** axios **from** 'axios'  
axios.defaults.baseURL = 'http://example.com/api/';  
Vue.prototype.$axios = axios;

添加完之后，就可以在 .vue 文件中调用该插件了：

**this**.$axios.get('managerStatus.action')  
 .then(**function** (response) {  
 **…**  
 }).catch(**function** (error) {

…  
 });

## 注意事项

* **尽量不要用JQUERY，如非必需情况，避免在script中直接操作dom**
* **不允许使用HTML4或以下的html标签，如<font>等**
* **尽量使用HTML5的语义化标签，如<header> <footer>等，代替<div class=’header’>**
* **尽量使用ES5/ES6的语法：**
  + 模板字符串
  + 解构
  + const/let
  + 箭头函数
  + Array.forEach, Array.indexOf等数组、对象的操作方法
  + promise链式调用

……

* **CSS尽量写在每个.vue文件的<style>标签中，并加上scoped**
* **可以使用SCSS代替CSS**
* **使用axios代替ajax**
* **遇到问题先找API，API找不到再百度**
* **CSDN鱼龙混杂，存在很多没有质量、抄袭、过时的文章**
* **前后端分离**
* **使用ESLint规范代码**
* **前端项目不要再用MyEclipse开发**
* **学会看报错信息，Vue的报错信息比ext, arcgis的有价值**
* **能用20行解决的问题不要写成200行**
* **各种命名，尽量不要用缩写**
* **能在iview/element中找到的组件，不要另行找插件，特别是jquery插件或者CSDN上别人自己写的插件**