# Vue的简单使用说明

1. **所需关注技术**

**必需掌握的基本技术：**

1. webpack
2. vue-resoure : 对接后台HTTP
3. vue-router: 路由
4. vuex: 状态管理，借鉴自react的redux
5. vue-touch 2.0 (移动端触摸事件，因为vue本身不提供touch事件)

API文档： <https://github.com/vuejs/vue-touch/tree/next>

（注意：低版本的vue-touch不支持vue 2.0）

1. **项目构建流程：**
2. 官网推荐vue-cli自动构建项目



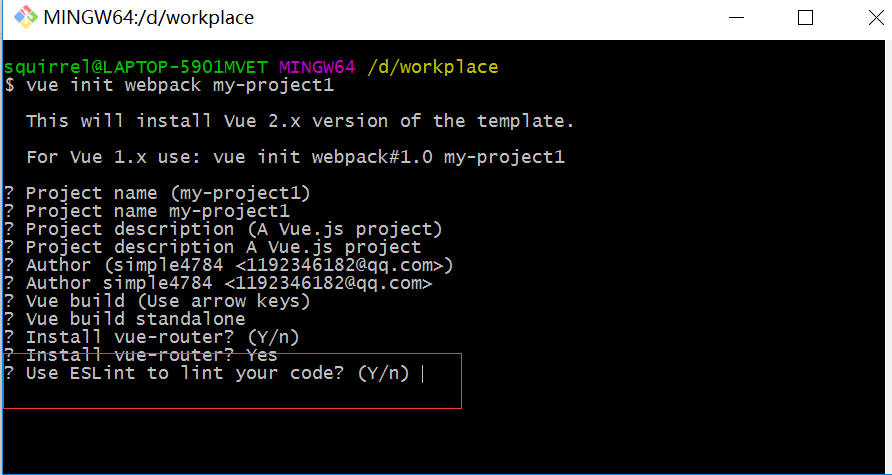
1. 安装相关依赖：

npm install vue-resource vue-scroller vuex --save

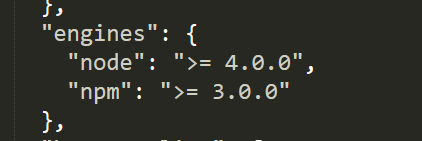
npm install vue-touch@next

1. 安装事项

1）个人习惯： 安装过程中，不选择ESlint进行语法检查，因为Eslint检查有点太过严格。。。



2）一般自动构建的项目，默认使用vue 2.0版本，对npm和node的要求是：（package.json）



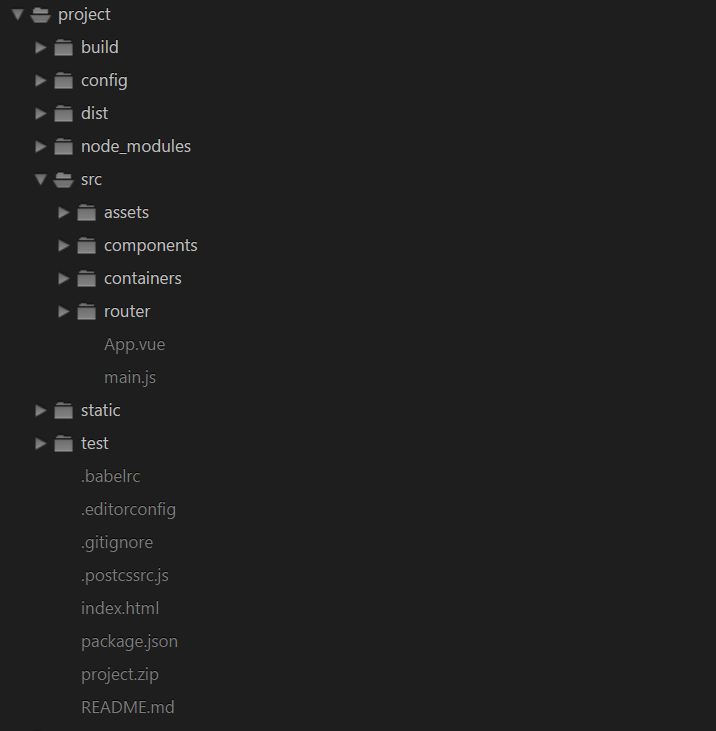
3） 推荐使用node版本控制工具，我的window系统使用nvm进行多个node版本的自由安装、切换、卸载等（安装nvm之前先把系统本身的node卸载）

4）使用淘宝镜像cnpm代替npm，安装依赖的速度更快。安装方法：

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

一般拿到别人项目，依赖安装： 先cnpm install, 后npm install

1. **文件目录介绍：**



（1）bulid: 项目构建的相关代码

（2）config: 开发环境配置

其中可在其目录下的ndex.js ：进行反向代理的设置，设置后台接口地址、跨域等。如图：



（3）src: 源码目录

assets: 资源目录

components: 子组件

containers: 父组件和页面

**router: 前端路由**

**main.js 程序入口文件 （公共样式，js等可在这里配置，粗略理解为可在这里配置全局的东西）**

static: 一般存放静态文件，比如一些图片，json数据等。

.babelrc： ES6 语法编译配置

.editorconfig： 定义代码格式

.gitignore： git上传需要忽略的文件格式

**index.html： 入口文件**

package.json： 项目基本信息

README.md： 项目说明

1. **以vue-touch为例，说明一般依赖模块的安装和使用过程（具体根据相应API说明进行使用或者查看模块的README.md说明）**
2. npm install XXX -save
3. main.js 文件进行模块的引入和配置



1. **组件间的通信和对比（因为我之前使用的是vue1.0, 对vue2.0版本不大熟悉）**

|  |  |  |
| --- | --- | --- |
| 版本 | Vue 1.0 | Vue 2.0 |
| 父组件传子组件 | Props | props |
| 子组件传父组件 | (1)A，B通过$dispatch调用APP的Event并传递数据给APP;  (2) A通过$dispatch调用APP的Event并传递数据，APP通过$broadcast调用B的Event并传递数据 | 子：$emit(eventName)  父$on(eventName) |
| 非父子组件的通信 |  | 建议使用vuex进行状态管理 |

注意：在vue2.0中$dispatch 和 $broadcast被弃用，因为基于组件树结构的事件流方式实在是让人难以理解，并且在组件结构扩展的过程中会变得越来越脆弱,并且这只适用于父子组件间的通信。即：

vue2.0 移除：1.$dispatch() 2.$broadcast() 3.events

1. **vue的生命周期**

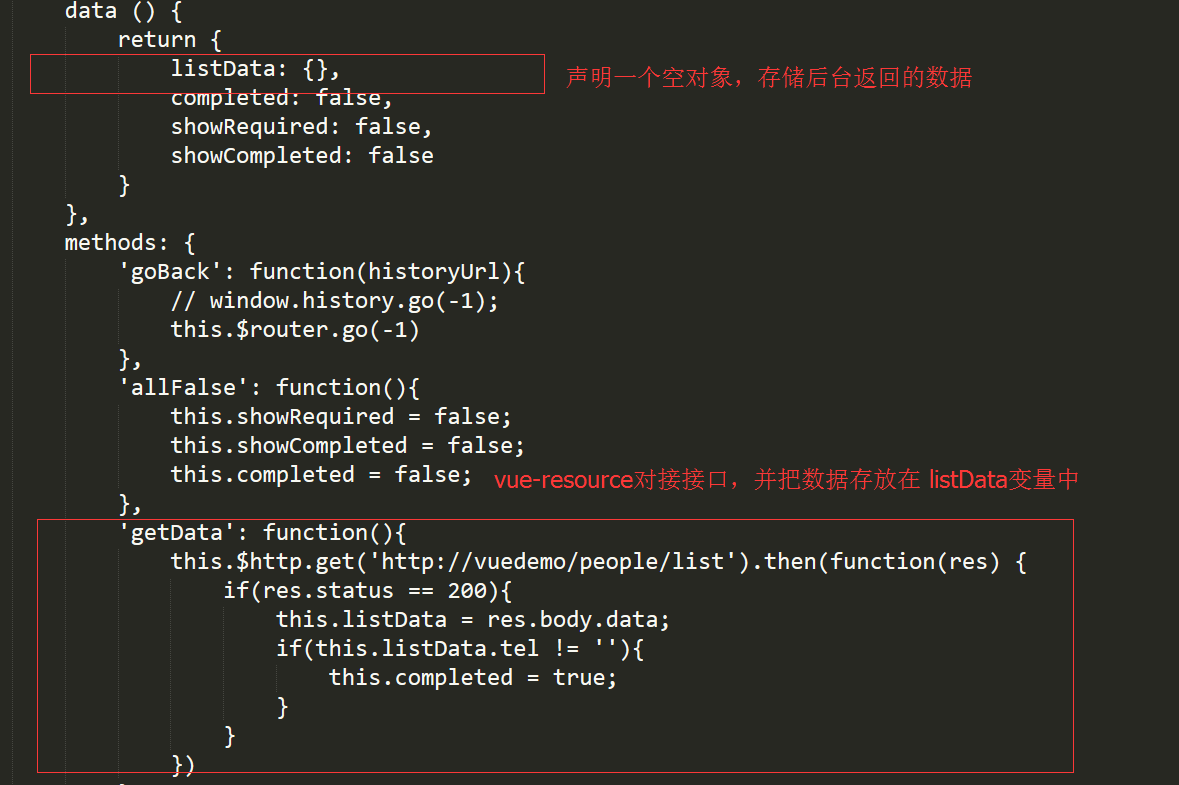


1. **ajax请求发起的阶段：**

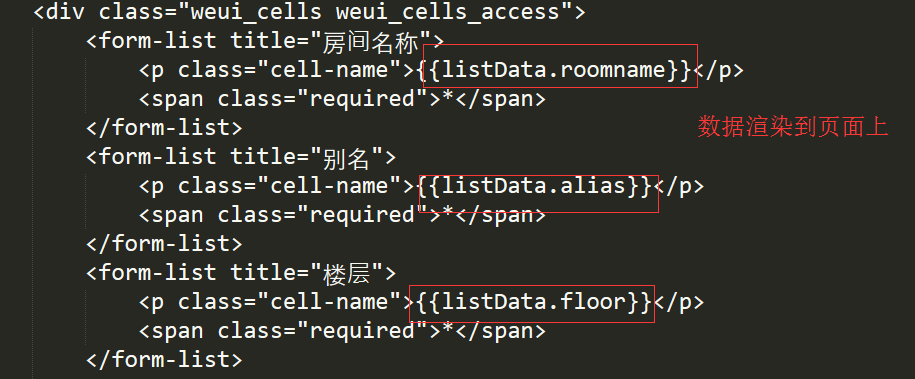


1. **vue-resource 简单使用以及后台数据渲染 （以people.vue为例）**

script:



template:



1. **vue-touch 方法的简单说明**：<http://www.jb51.net/article/111805.htm>
2. 举例说明组件的划分（组件划分很重要，切勿划分过大！！！！！！）

说明：子组件的数据 由 父组件通过props传递



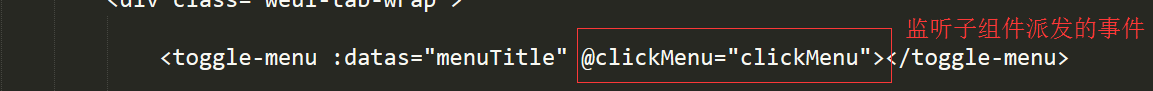
**以组件 3 为例：**

子组件（toggleMenu.vue）：



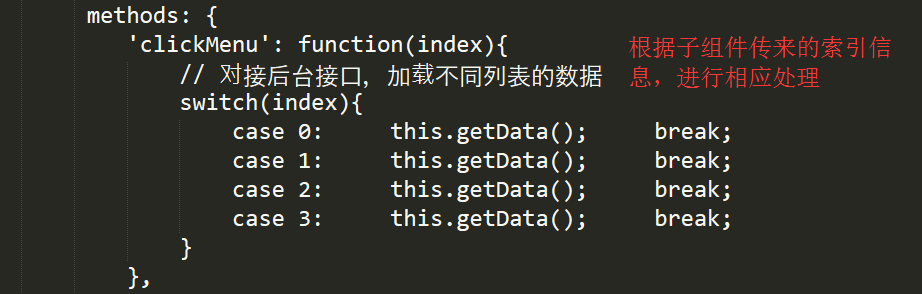
父组件(house.vue)：

Template:



Script:

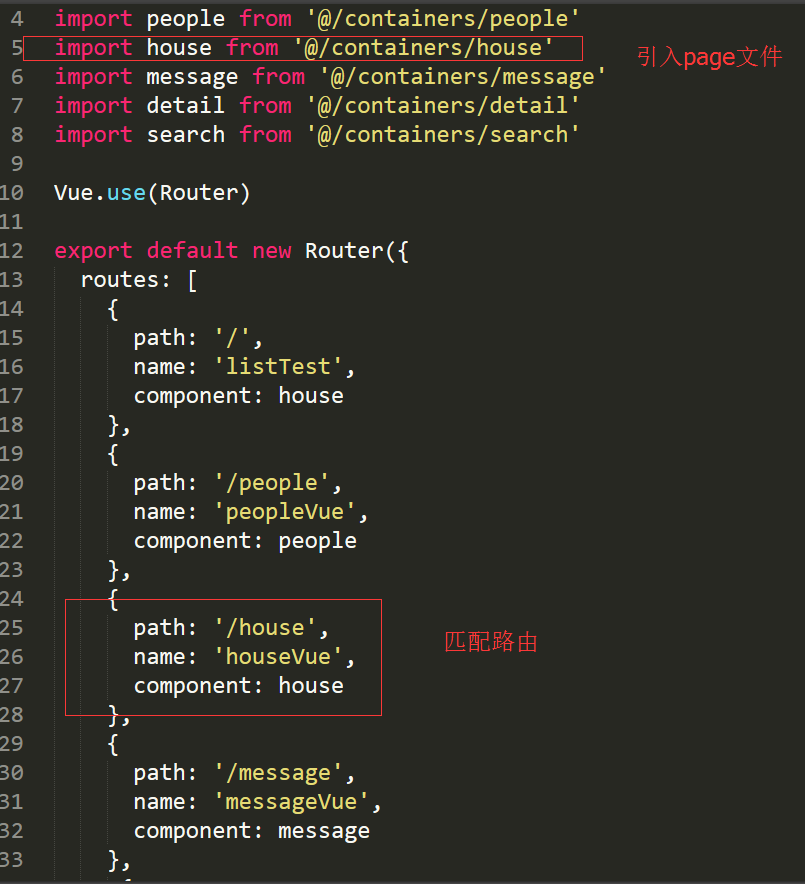




1. **页面显示流程（参考上一点）**
2. 子组件的编写(如：在components编写toggleMenu.vue)
3. 多个父组件构成单个页面

在containers引入子组件编写父组件，并构成单页面

1. 路由的设置（route下的index.js进行设置）

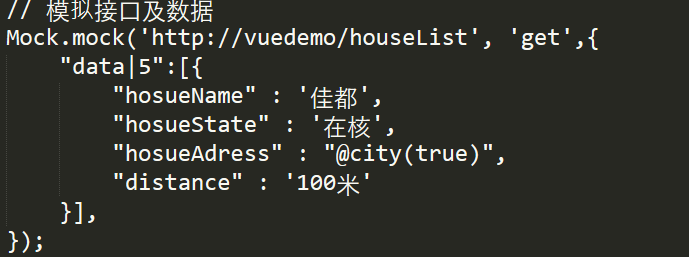


1. **从入口文件开始看项目**
2. 入口文件： index.html main.js
3. App.vue 文件: 根元件，其他所有组件都是挂载在它上面
4. route/index.js: 从路由开始查看单个页面
5. containers 下的文件

（5）components 下的子组件

**13、可忽略：**

截图中，使用mockjs进行后台的接口和数据的模拟，以便更真实介绍vue-resource的使用,可忽略。



说明: 由于我之前用的都是vue1.0版本且使用时间比较短，没用vue开发过移动端页面，所以如果文档整理不好请多包含。