### 官方文档 vue;vue-router;vue-resource

<https://cn.vuejs.org/v2/guide/>

Vue实例中所有API接口

https://cn.vuejs.org/v2/api/#el

<https://router.vuejs.org/zh-cn/> 路由功能

<https://github.com/pagekit/vue-resource> 前后台数据传输

<http://vuex.vuejs.org/zh-cn/> 跨组件之间传输数据

是一套构建用户界面的**渐进式框架。**

### Vue与node.js、npm、webpack的关系

<https://segmentfault.com/q/1010000004460077>

<https://segmentfault.com/q/1010000008451764>

http://www.runoob.com/nodejs/nodejs-npm.html

node.js是一种服务端脚本语言。Npm作为node的一部分，主要作为包管理软件，用于自动下载、应用发布在网络上的开源node.js库文件等资源。这样是很多VUE安装教程会有相关的NPM安装内容。

1， npm install是帮助安装 vue,或React到本地，npm install也可以安装vue,React的开发工具。当然你完全可以像jQuery一样自己找网站下载下来，在页面中引入。  
2,没有npm可以使用vue，vue只是一个前端框架。  
3，4，**node是js的服务执行环境，前端常常借助 node帮助前端实现工程化**。很多工具是基于node的，比如说你说的webpack,grunt。  
5，webpack,grunt是前端工程化的工具，可以帮助前端完成混淆压缩，校验等工作。1

### Vue中component组件是什么意思

<http://www.tuicool.com/articles/Qze2my2>

根据上午，简单讲就是：基于html基本元素与VUE的js语法，实现的自定义标签。这些标签绑定了js事件。其中，App.vue是主组件，在其中可以引入，使用componen目录下vue后缀的文件所代表的组件。组件一般包括<templete>、<script>、<style>三节，分别代表html、JS、css.

### 创建Vue项目

<https://segmentfault.com/a/1190000007441374>

通过vue-cli客户端创建VUE项目，然后可以在IDE环境中编辑。

<http://www.jianshu.com/p/0f40d627a39b>

<https://www.zybuluo.com/xudongh/note/758521>

上述是一个比较完善的教程。

### [vue.js项目main.js、App.vue、page/index.vue的引用调用关系是怎样的](https://segmentfault.com/q/1010000008863946)

<https://segmentfault.com/q/1010000008863946>

### Vue主要部分、详细创建过程、具体实例

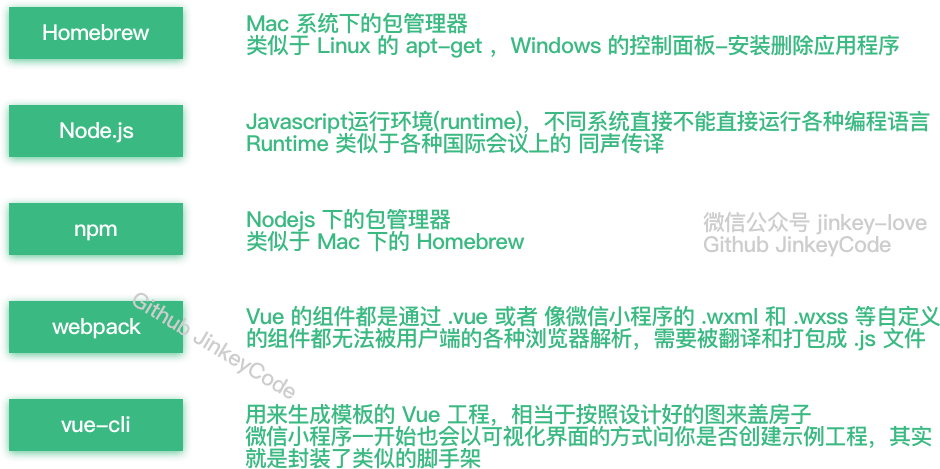
<https://segmentfault.com/a/1190000007328936>

### WebStorm中调试运行vue.JS项目

<http://www.jianshu.com/p/0f40d627a39b>

### Vue安装、demol教程

http://www.open-open.com/lib/view/open1476240930270.html



### 菜鸟主页的VUE“新手概览教程”——很不错

<http://www.runoob.com/w3cnote/vue2-start-coding.html>

### Npm的使用教程

<http://www.cnblogs.com/chyingp/p/npm.html>

既然node.js是各种JavaScript的运行环境，NPM作为包管理软件，自然是管理这一平台下的诸多软件、组件。如webpack、vue-cli脚手架，vue、vue-resource等modual组件。上述链接主要对npm的使用作一定解释，**如本地安全和全局的安全的实现方式与区别。如何安装、删除。**

### Zyf推荐的vue入门

<https://segmentfault.com/a/1190000007124470>

<http://aevit.xyz/2016/06/29/vue-in-laravel-1/>

element的环境：http://element.eleme.io/#/zh-CN/component/form

### 项目环境

<https://gitlab.lvqu.me/admin/users/zhangkailong>

123456789 默认密码

### PHPStorm中开发Lavaral项目

<http://www.siguoya.name/pc/home/article/239>

Lavaral是一个PHP框架，主要用于开发PHP Web Server。

### Webpack基本学习

Webpack有一个很重要的配置文件，webpack.config.js。

<http://www.cnblogs.com/zuozuo-blog/p/6560485.html>

### Vue-router基础学习

基于vue.js 1.0的教程，解释很详细

<http://www.cnblogs.com/keepfool/p/5690366.html>

比较简洁的实现Vue路由功能（基于vue 2.0）：

<http://blog.csdn.net/sinat_17775997/article/details/54710420>

zyf发的router相关的内容：<https://router.vuejs.org/zh-cn/>

### New vue()与export default的区别

<http://www.codes51.com/itwd/4296450.html>

前者是创建一个vue实例，后者基于js 6语法导出moduel，以便于父组件import导入使用。详细区别可参考相关超链接。  
完全不是同一回事.  
前者是es6的module中的语法,后者是创建一个vue实例.  
我想引起你的误解是因为你用了webpack开发vue同时用了vue-loader.  
其实这最终会通过vue-loader处理成原生js代码.即无论如何都要new Vue()

### 基本指令

* V-bind 给element动态绑定属性值：将vue中data绑定至title属性；绑定class、style等属性值。
* V-if 依据对应布尔值，确定是否显示对应渲染元素
* V-for 循环输出数据
* V-model 将用户输入“双向”绑定至vue实例中data

### 通过laravel等框架实现前后端交互的路由（vue-resource）

在vue.js中vue-resource插件主要用于与后台的交互。例如以下是典型的用法

**this**.$http.get('/api/order-stat/booking-count',{  
 params:params  
}).then(**function**(response)

当然也可以如下：

**var** resource = **this**.$resource(**this**.apiUrl)  
vm = **this**resource.get()  
 .then((response) => {  
 vm.$set('gridData', response.data)  
 })

通过vue-resource的$http.get（）等方法向服务端特定的URI提交请求，并获得对应的response。

此处需要着重理解URI参数，（get方法一般用户获取特定页面资源，而post方法一般用户提交用户表单信息），此URI参数一般不会是服务端具体的PHP文件，而是“这个是api 接口地址，是lavaral 进行路由管理的，地址上并不会直接暴露接口方法所在的文件”。

可以简答的认为前端有路由，后端也有路由，laravel的理由功能可以参考：<https://laravel-china.org/topics/686/laravel-learning-notes-routing-basic>。

此外，还需要理解下RESTful架构，全称为：representational state transfer。详细参考如下：

<http://www.ruanyifeng.com/blog/2011/09/restful.html>

### 遗留问题

1. 表单空间（form或input）中，通过v-model将实例数据绑定到动态value上。详见vue官方课程：基础课程-表单空间绑定-value绑定。
2. 现在有多种不同的语法，比如export defaul {}和new vue{}两种vue的写法；注册组件时，有在vue实例中直接使用components：{}和创建实例前，使用vue.component（component-name）两种方法。  
   后续我们选择哪种方案？何故？
3. 什么是字符串模板？什么是DOM模板？相关：组件-prop
4. 父组件向子组件传输数据时，如何通过“非props”属性直接传输数据。相关：组件-非props属性
5. 除了export fault的方法，new vue使用方法的语法是什么？
6. Render函数之“基础”内容，将html内容直接写在<script></script>中。

### 实例、挂载、MVVM等基本概念

详细参考; <http://www.jianshu.com/p/56ba7488a4af>

区别于传统MVC架构，vue采用MVVM架构理念（事件触发、模型、界面之间相互影响，没有C）。对于单页面应用，只会new 一个vue实例。而vue实例通过el指令，将对应实例挂载到对应的id的元素。

### Mockjs、mock Server、json server等概念

起源于从github上下载一个vue练手项目（vueProject--Mall），这个项目不仅用到了vue常见的组件如vueRouter、vueResource、vueX等；还使用用了mockjs，作为前后端分离的情况下，vue项目运行的容器，另外该项目中还使用到了json文件，所以还需要通过“npm install –g json-server”安装json服务器。

关于mock.js，详细参考：<http://www.jianshu.com/p/d7c96bac6e5f>

Json-server主要负责数据这一块，详细参考：<http://blog.csdn.net/CSqingchen/article/details/48495353>，安装支持，可以直接在浏览器中访问读取访问相关数据。

[json-server快速“伪造”后台接口](http://blog.csdn.net/liqianglai/article/details/53488091)

json是一种轻量级的前台（js）与服务端数据交换的数据格式。通过相关接口，可以轻松实现json数据与JavaScript中数据变量的转换。可参考：《JavaScript高级程序设计》

### 问题记录

* IDE的语法报错问题，如下述方式中添加内容，会导致整个项目无法运行，即：所有页面都是空白、无法打开，（从主页面到各个路由页面），且IDE也不会报错。

**export default**{  
 data(){  
 **return**{  
 test:'123'  
 }  
 },  
 }

components（‘anchoredheading’）

* php artisan serve是运行这个OMS项目，包括前后端；Npm Run dev应该是node.js环境，模拟运行前端页面。是否正常前端开发过程中，只执行后者便行了。但是，我试了貌似不行，必须要运行php；

zyf的解答：第一个是代码运行环境，就是php环境，是必须要的，npm run是代码编译工具，如果不调试代码可以不用运行。我们的项目使用laravel来管理路由，相对于纯粹的vue项目，可以认为laravel包含了vue。我们是使用laravel来编译发布的。

* node socket.js 可以让项目在调试过程中，浏览器控制台出现refuse报错。

### 父子组件之间传输数据

父组件向子组件传输数据通过：props属性实现；

1. 在子组件总直接定义props属性
2. 在父组件使用子组件标签时，直接使用v-bind：propsPara=value，将对应的子组件常量或变量传递至子组件；
3. 在子组件中，可以通过compute或data或method等形式二次处理props参数。

子组件向父组件传输参数：

1. 在子组件中定义事件，在事件触发方法中，通过this.$emit(fatherFunction,para)将对应参数提交至父组件的对应方法中。
2. 与此同时，父组件中使用子组件标签时，应当使用<component v-on:childFunction=’fatherFunction’></component>，详细代码参考如下：
3. <template>  
    <div id="renderFunctionChild">  
    <button v-on:click="clickMe">12234</button>  
    </div>  
   </template>  
     
   <script>  
    **export default**{  
    data(){  
    **return**{  
    childTest:'qbc'  
    }  
    },  
     
    methods:{  
    clickMe:**function** () {  
    **this**.$emit('child-say',**this**.childTest)  
    }  
    }  
    }  
     
   </script>

<template>  
 <div id="renderFunction">  
  
 <!--父组件向子组件传递数据-->  
 <anchoredheading v-on:child-say="getParaFromChild"></anchoredheading>  
 <p>{{test}}</p>  
 </div>  
</template>  
  
<script>  
 **import** anchoredheading from './renderFunctionChild.vue'  
 **export default**{  
 data(){  
 **return**{  
 test:'123'  
 }  
 },  
   
 methods:{  
 getParaFromChild:**function** (childTest) {  
 **this**.test=childTest  
 }   
 },  
 components:{anchoredheading}  
 }  
</script>

### OMS项目调试

1. 本地安装PHP环境，并配置对应的path环境变量
2. 利用php composer.phar update --no-scripts 更新安装对应的php依赖包（update依赖于composer.json文件）；如果在本地直接安装了composer并配置了相关的 环境变量，则，直接运行composer install/update –no-scripts.
3. php artisan serve启动项目

### 通过scope属性，在子组件和父组件之间通过作用域插槽传递属性

<http://www.jianshu.com/p/a0a83029a217>