1.缓存

在<router-view></router-view>包一层<keep-alive>，用来缓存  
  
  
2.子向父通信   
 （1）子组件触发事件，  
   
 has-log后面的data作为参数传回父组件

（2）父组件执行回调函数onSuccessLog()  
   
 

1. 引入外部文件  
    （1）CDN可以放在在index.html中  
    

（2）本地JS放在script中 import方式  
 本地CSS放在style中引入



（3）如果不能用CDN也不能本地引用，比如font-awesome字体库，在打包后的文件中，把字体库文件夹放在CSS中，打包后的index替换相应路径即可。  
 本体引入字体库和CDN引入不同，CDN的CSS文件里还会从网上再引入font字体库

  
4.当在命令中引入带符号 - 的类名时，把该类用单引号包起来即可  
  
5.methods中的函数使用data里的数据时，要加this this.data  
 打印data中的数据也要加this  
 方法互调也要加this

6.v-for子列表this作用域

通过this改变子项的样式  
 （1）当只生成一个列表的时候，可通过index解决  
 单选：  
62`URDM59B~}WVW7LSP6CO2  
]@ILNMHJ`CFLLA1$US`4T5Y  
  
 多选：  
 首先创建个空数组，然后把index传给子项的回调函数，回调函数把这个index值和空数组进行比较，没有就存，有就删；再通过第二个回调函数判断index在不在index中，在就改变样式不在不改变等，要导入lodash库  


（2）当生成多个平级列表的时候，此时不传index，传元数据的id值即可  
  


7.<router-link>实现a标签伪类active特效  
 在标签中加一个 active-class 即可。Hover伪类不需要模仿，所有元素都可用。

After伪类通过替换父类名来更换  


8.JS操作标签的属性，操作CSS看第十条  
 使用v-bind  
 





9.npm run build 生产环境

npm run build生成的代码 在服务端用httpserver直接访问即可,或者Nginx代理指向dist目录,如果是本地(自己电脑)看效果,需要搭建个HTML服务器进行访问,或者用脚手架自带的,执行npm run dev也可以

默认生成的文件仍需要node启动下  
当要生成静态文件的时候，要更改config里的index.js



引入的CDN要加http，或者build后把cdn文件放进CSS中再引入  
CSS引入的文件要安装下面方式更换  
 

10.CSS样式height的值绑定浏览器高度  
   
  
11.在JQ项目中使用VUE  
 (1) 直接在子页面使用的话，会报错vue is not defined  
 在首页或者子页的父页引入VUE，子页面就可以直接用了  
如果首页也想用vue，可以首页外再嵌套一个页面，父页面引入VUE

(2)JQ默认不能获得VUE创建的动态节点，所以ready函数会失效  
解决办法： <http://www.jb51.net/article/87287.htm>  
<http://www.w3school.com.cn/jquery/event_live.asp>  
 把需要节点加载完调用的函数放在on中执行  


12.VUE单元格合并  
 需求：把1变成2  
  
  
  
第一种情况：后台传json没有rowspan  
 <http://www.cnblogs.com/xianrongbin/archive/2012/10/05/2712624.html>

其中，tabObj代表表的Id，即GridView的Id，cellIndex代表合并第几列，beignRow 代表从几行开始，通常使用1，因为0是表头哈。

js代码如下：

function SpanGrid(tabObj, cellindex, beginRow) {  
    var colIndex = cellindex;  
    var rowBeginIndex = beginRow;  
    if (tabObj != null) {  
        var i, j, m;

        var intSpan;  
        var strTemp;  
        m = 0;  
        for (i = rowBeginIndex; i < tabObj.rows.length; i++) {  
            intSpan = 1;  
            m++;  
            strTemp = tabObj.rows[i].cells[colIndex].innerText;  
            for (j = i + 1; j < tabObj.rows.length; j++) {  
                if (strTemp == tabObj.rows[j].cells[colIndex].innerText) {  
                    intSpan++;  
                    tabObj.rows[i].cells[colIndex].rowSpan = intSpan;  
                    tabObj.rows[j].cells[colIndex].style.display = "none";  
                }  
                else {  
                    break;  
                }  
            }

        }  
        i = j - 1;  
    }  
}

第二种情况 后台传的json有rowspan值,只合并第一列的单元格

如果没有rowspan 可以循环加个  
 思路：后一行的机场名和前一行的机场名相比，相同则在该json对象中改rowspan的值为false，第一个td渲染会判断rowspan的值，为真渲染  
https://github.com/bill-mark/shandongAir/tree/master/%E9%9D%99%E6%80%81%E4%BB%A3%E7%A0%81


//合并单元格算法tabObj代表表的Id，即GridView的Id，cellIndex代表合并第几列，beignRow

代表从几行开始，通常使用1，因为0是表头

这个方法要放在update中使用

const SpanGrid = function(tabObj, cellindex, beginRow) {

var colIndex = cellindex;

var rowBeginIndex = beginRow;

if (tabObj != null) {

var i, j, m;

var intSpan;

var strTemp;

m = 0;

for (i = rowBeginIndex; i < tabObj.rows.length; i++) {

intSpan = 1;

m++;

strTemp = tabObj.rows[i].cells[colIndex].innerText;

for (j = i + 1; j < tabObj.rows.length; j++) {

if (strTemp == tabObj.rows[j].cells[colIndex].innerText) {

intSpan++;

tabObj.rows[i].cells[colIndex].rowSpan = intSpan;

tabObj.rows[j].cells[colIndex].style.display = "none";

}

else {

break;

}

}

}

i = j - 1;

}

}

13.在VUE项目中使用JQ

https://cn.vuejs.org/v2/examples/select2.html  
 先配置  
 <https://segmentfault.com/q/1010000005596299>

  
 再使用  
   
  
  
  
  
14.VUE绑定具体的内联样式  
   
 ：<div v-bind:style="{ color: activeColor, fontSize: fontSize + 'px' }"></div>

当绑定图片URL的时候，直接写成 x="url(./css/images/baowenDown.png)"

url里面不用再加双引号，vue会自动加上

15.(1) v-for 根据json里对象的不同属性值设置class

思路：根据json对象不同的值改变某一个属性为布尔值或新增一个属性，模板中class名再绑定到这个属性值上



效果：如果ISHAVECHANGE=0，则添加一个样式类dark-line

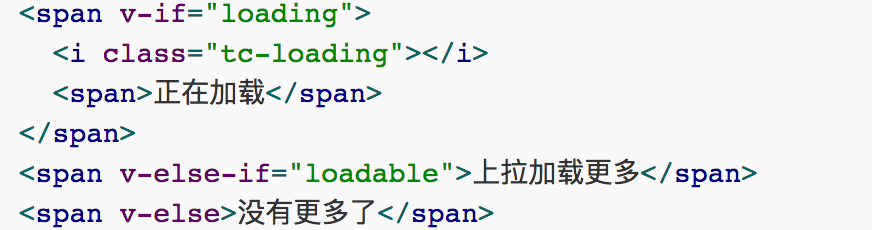
15.(2)根据tem的位置设置样式

<https://segmentfault.com/q/1010000007589795?_ea=1392912>  
 

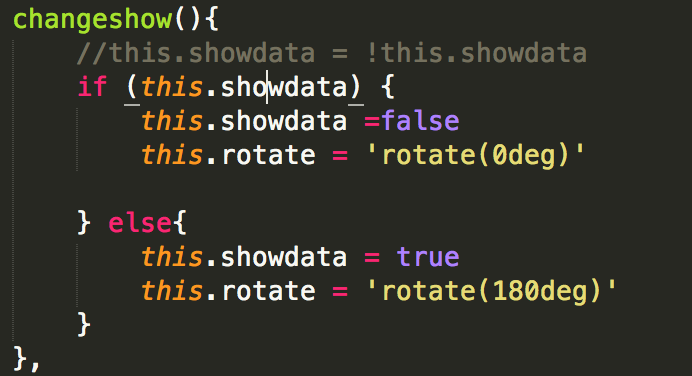
这样生成的列表会多一个类 比如第三个列表多个item2 样式类

15.（3）根据index设置样式，传index给函数，函数再判断真假 三元选择

 也可以用v-if 写两个div 一个 v-if=”data” 另一个 v-if=”!data”

或者用v-else-if  


或 变量就是这个值,if(data){}else{}



16.methods里调用mounted创建的对象  
 如果调用mounted使用的方法，先在methods里声明这个方法，再在mounted里调用这个方法，然后methods里其他函数也可以调用这个方法了。  
 如果调用mounted里创建的对象，现在data里声明这个对象，再在mounted里修改这个对象，methods里就可以对这个对象进行修改了。  
   
   
  
  
17. 测试build后的文件

https://www.cnblogs.com/ang-/archive/2017/06/26/7082202.html

build成功后有个tip提示你build后的文件需要部署在http服务器上，不能通过file协议打开。

我们可以通过node-static来启动服务。可以写一个js配置文件通过node来启动，或者CLI中输入static dist(先安装node-static)：

$ static dist

serving "dist" at http://127.0.0.1:8080



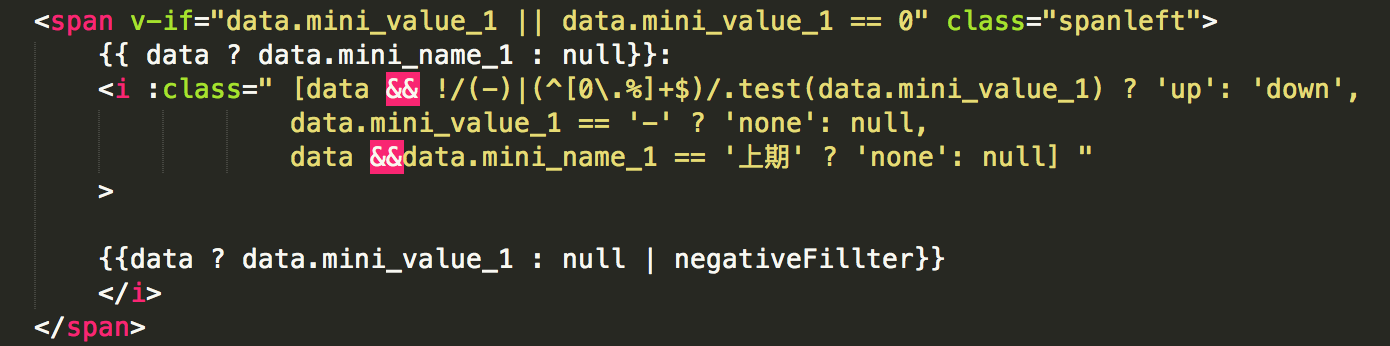
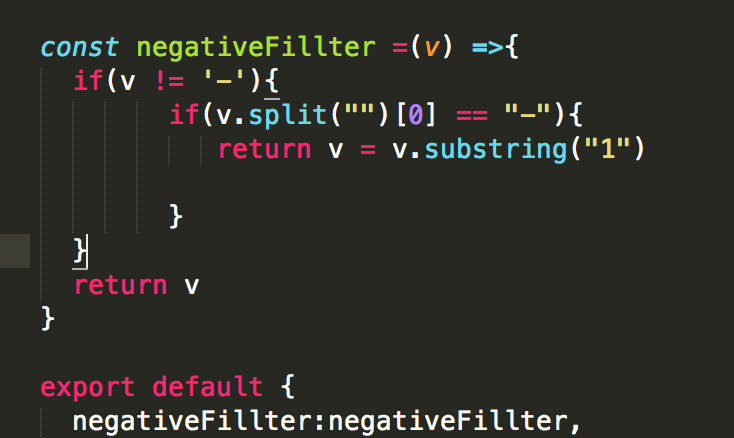
第一种方法:CSS子元素选择器  
<http://www.qdker.com/archives/103.html>

第二种方法项目地址https://github.com/bill-mark/aasian-ams  
第一步：偶数行list加一个属性  
  
第二步：添加个响应属性  
  
  
18.Vue cli项目获得函数回调函数的返回值  
<https://segmentfault.com/q/1010000004237026?_ea=539265>  
  
 把回调函数写成箭头函数的形式即可或者this起别名  


19.VUE+正则+less 判断数的大小并依此响应样式

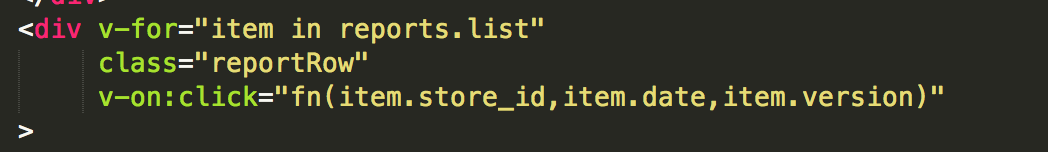
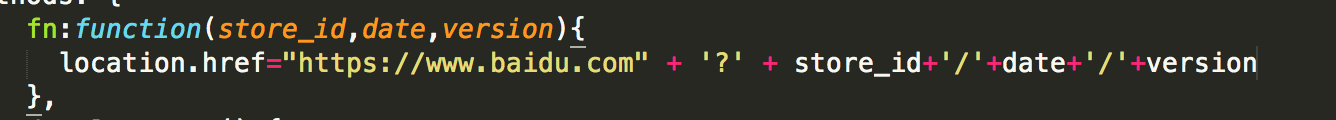
当数大于0 ,classname的名为up,否则为down  
http://www.runoob.com/regexp/regexp-tutorial.html  
   


20.v-if的使用  
 当两个div的状态依赖同一个变量的时候.可以一个依赖正常状态,一个依赖取反状态  
 <div v-if=”date”></div>  
 <div v-if=”!data”></div>

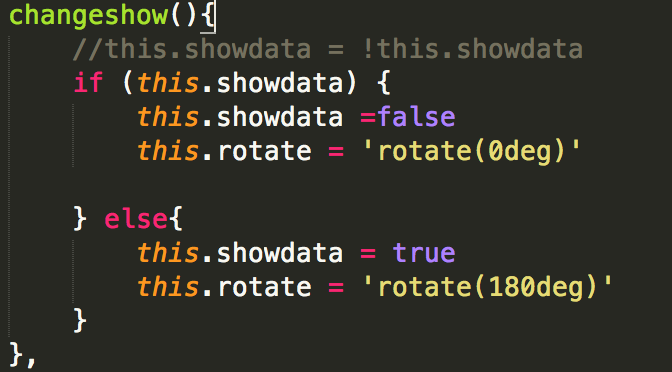
21.生命周期:updated  
 当组件接收到props在执行某个函数时,要用到updated  
 或者获得通过props传过来的值,在updated里获得即可,当watch不好用的时候可用这个方法  
22.过滤器   
 需求 变量有无进行判断,有的情况如果变量为负数变为正数  
 解决方案:三元选择器和过滤器配合使用, negativeFillter是个筛选函数  
  


23.生命周期 created和mounted

如果两个函数先后操作一个双向绑定的数据,先操作的函数放在created里,后操作的放在mounted里

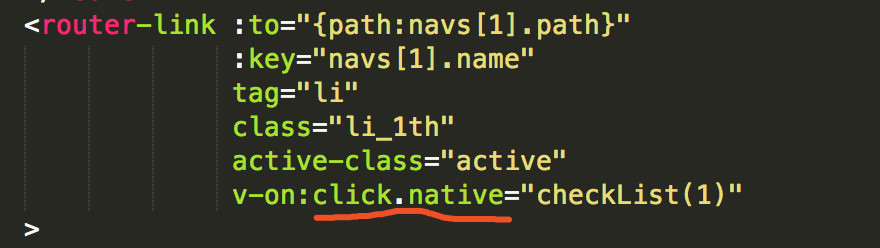
24.v-for 生成的HTML 带参数跳转外链  
 方法:点击执行一个函数,把参数动态传给函数,在函数里跳转  
  
  
再用var getInfo=window.location.search.slice(window.location.search.lastIndexOf("?")+1) 拿到参数  
如果参数地址前有# 则拿不到参数,比如  
<http://localhost:8080/#/?store_id=ADBJ37&week_no=201730&bversion=20170810> 拿不到地址

25.v-on  
如果想在v-on里绑定多个函数,可以先在v-on绑定一个函数,在这个函数中再通过this调用其他函数

26.v-show 和 v-if  
 需求:点击元素取正 再点击元素取反 循环  
 方案1: this.showdata = !this.showdata  
 方案2:  


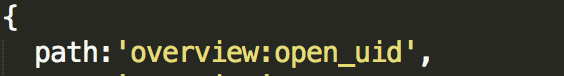
27.vue-router排bug Missing required prop: "to"  
 当router-link标签里没有to的时候会提示这个错误

28.绑定this  
 问题:一个函数里包含一个回调函数,如果这个回调函数想调用父函数同级函数,使用this调会提示not a function,正确的方式是在父函数中绑定this到一个变量上比如\_that再调用

29.在vue-router中绑定click事件  
 原理:给组件绑定原生事件  


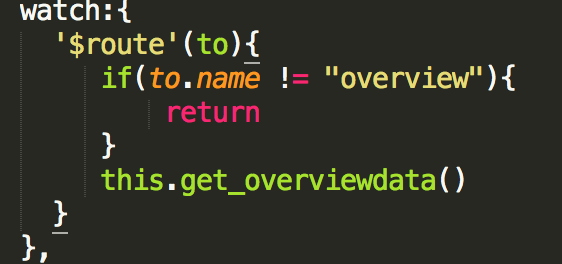
30.vue-router 传参数  
 传:路由出口不要加keep-live不然参数不更新;当用params的时候只能和name搭配用

当用name跳转带参数的时候,需要在路由配置处加上参数名,比如

  
 接: 目标页面 mounted this.$route.params

如果路由不变,仅参数变化,要响应参数变话,需要watch router,同时判断下路由跳转目标,相同执行ajax更新函数,或者用beforeRouteUpdate

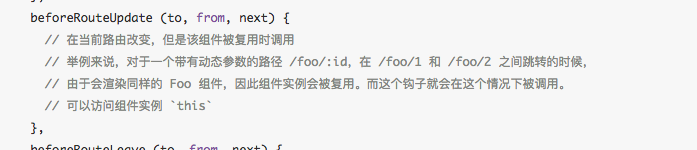
注意:当使用懒加载时,同级跳转检测不到



<https://router.vuejs.org/zh-cn/essentials/dynamic-matching.html>

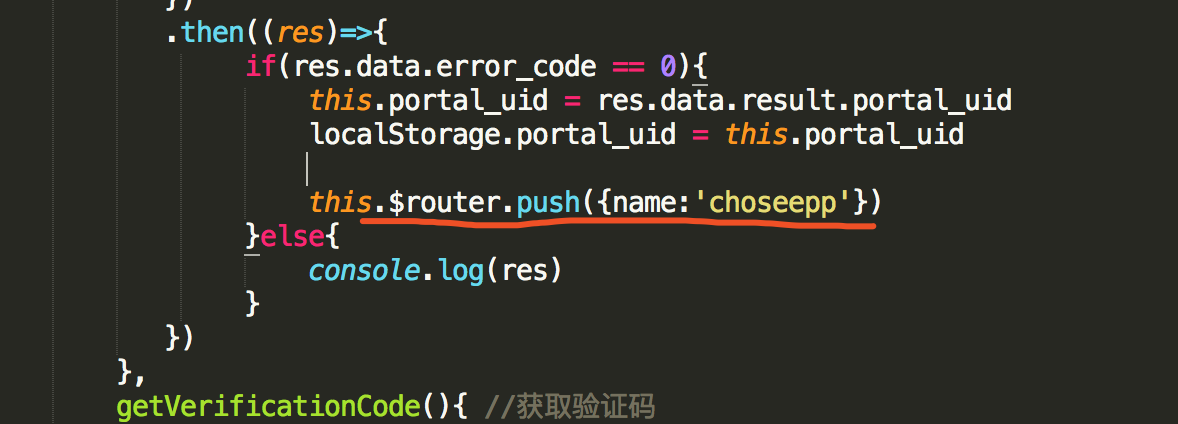
或者使用路由守卫.根据参数修改数据

<https://router.vuejs.org/zh-cn/advanced/navigation-guards.html>



31.Vue路由函数跳转  
 this.$router.push({name:'overview'})

如果在ajax成功后要执行函数,函数跳转的时候要放在ajax then后面跳转

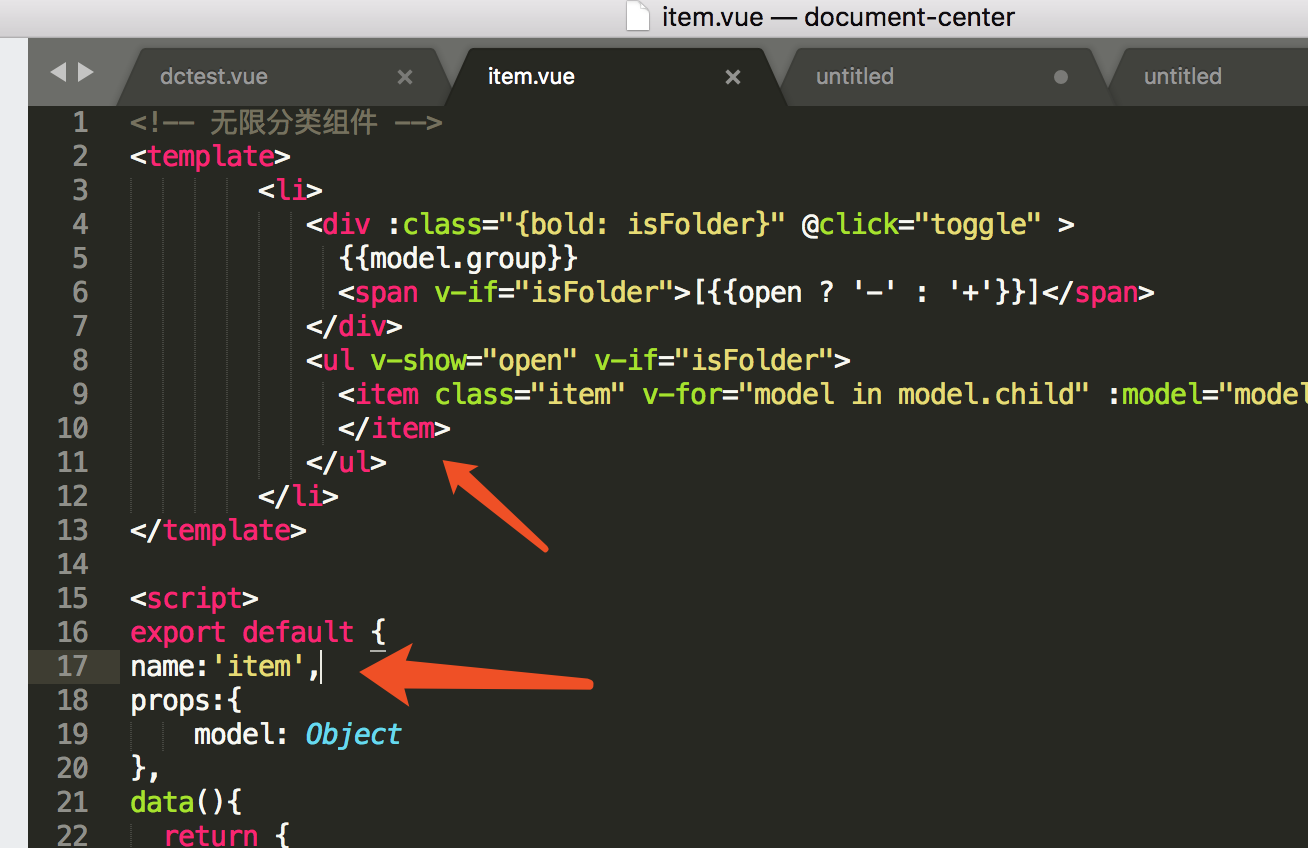


32.过滤器  
 使用场景:API返回1 页面中要显示已上线

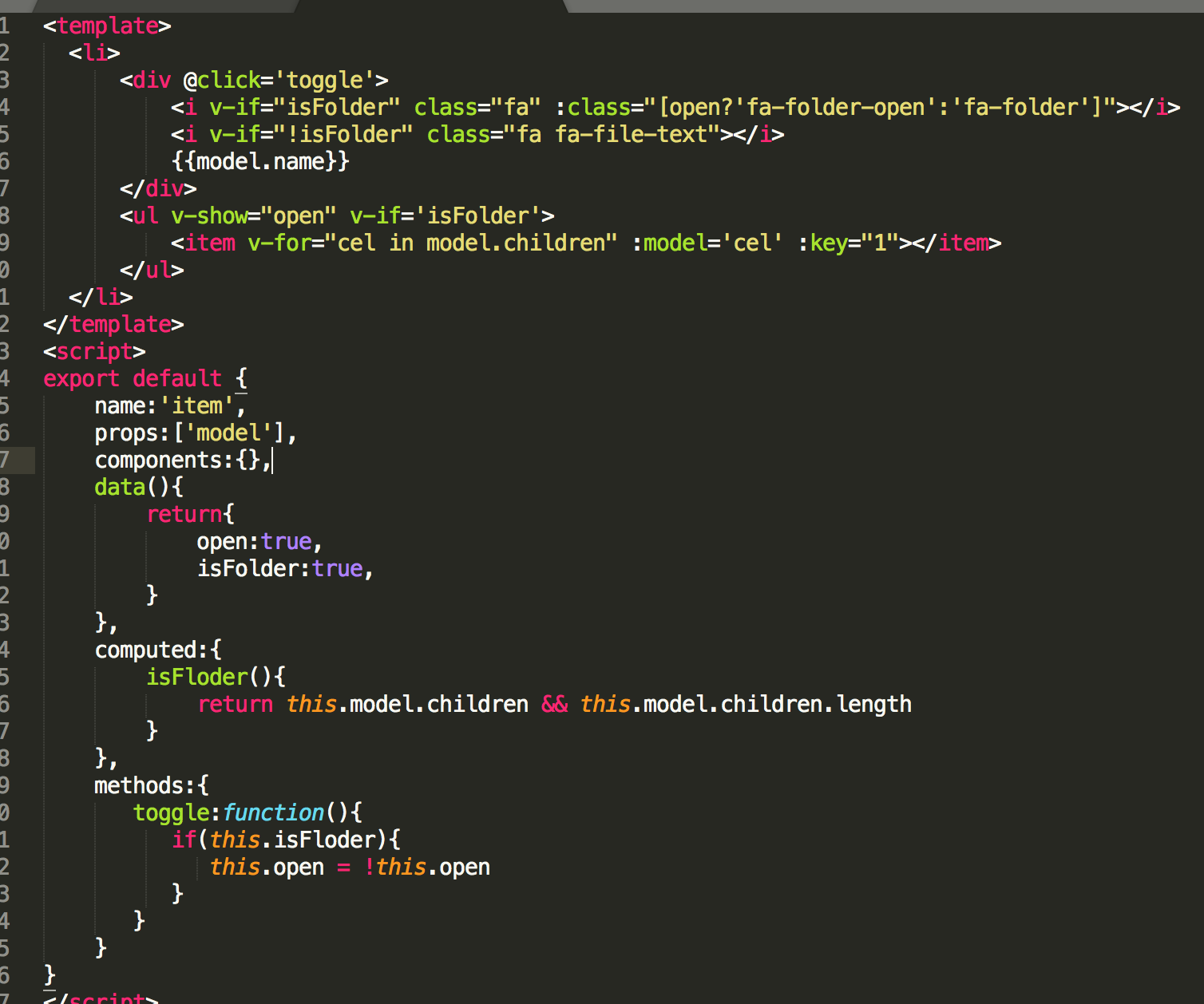
或者和v-for搭配用  
 如果直接使用函数写法比较麻烦,使用过滤器简单

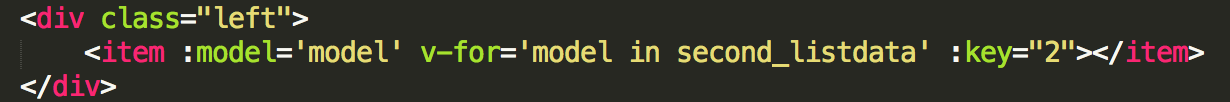
33.组件调用自身(递归)

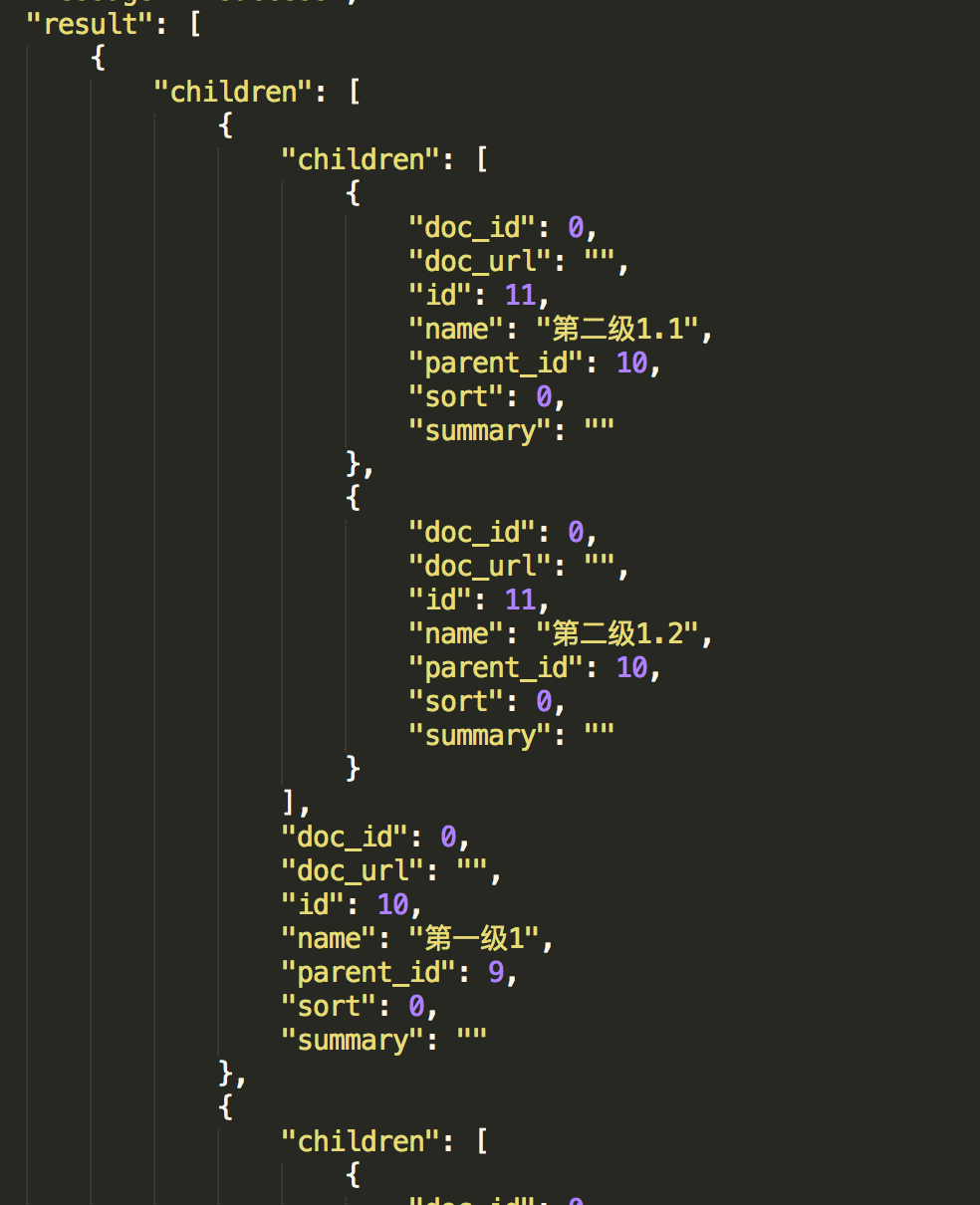
需要name声明下



递归tree:  
 <https://www.cnblogs.com/ychl/p/6075106.html>  
 思想:计算属性判断是否还有子节点,有就把子节点值传给组件自己,v-FOR的形式



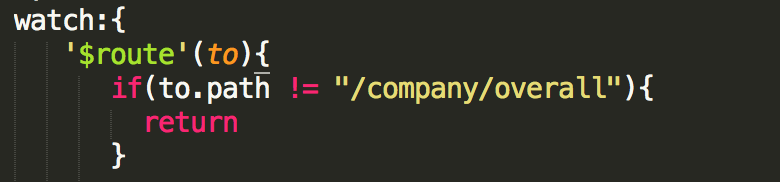
父组件引入  
 



34.element UI  
 (1)修改样式  
 先去掉style的scoped,再进行覆盖修改既可以,要注意样式污染问题 最好借用sass嵌套在一起 .a{ .b{} }

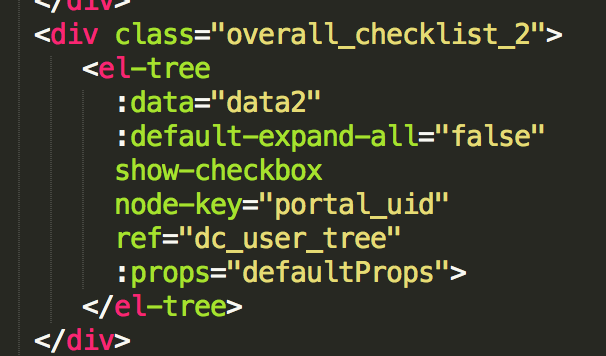
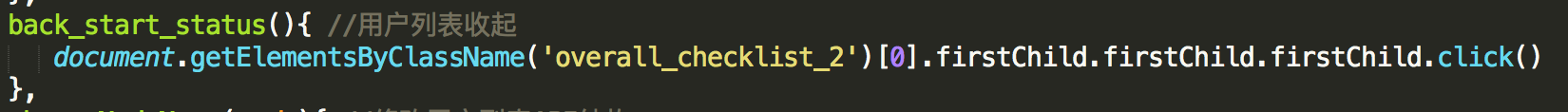
(2)按需引入的时候,由于element使用ES6标准 需要装个插件  
 npm install babel-preset-es2015 --save-dev

35.vue更新视图  
 第一种:路由传参数,获取参数刷新页面  
 第二种:watch路由路径或别名是否为当前路径,为true先重置data里的数据再执行一遍mounted里的方法(只更新数据,不刷新页面,推荐)



第三种: watch路由路径或别名是否为当前路径,为tru执行window.location.reload() (优点不用再重置数据写一遍mounted函数了 缺点:会刷新页面)

第四种:VUEX (推荐)

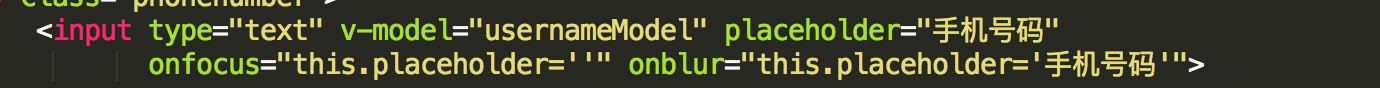
36.element ui  
 当使用tree组件的时候,如果需要点击别的地方收起tree,方法:  
 获取所在节点,执行click函数  
   
 

37.点击其他地方关闭弹窗  
 第一种:有阴影给阴影处加个函数  
 第二种:无阴影,监听全局点击事件(最外层DIV加个ID,监听这个ID点击事件),目标元素再增加事件冒泡阻止,最后页面销毁的时候移除监听事件  
 <http://blog.csdn.net/luckylqh/article/details/52860246>

<https://segmentfault.com/q/1010000008933630/a-1020000008933793>



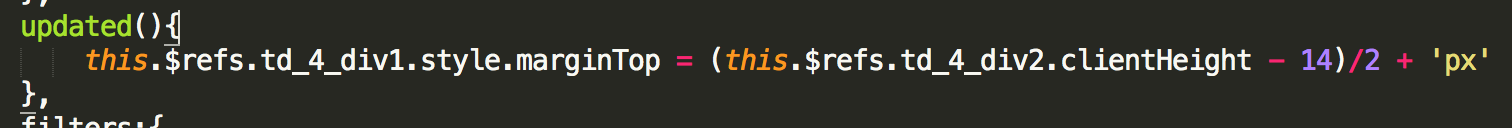
38.输入框placeholder点击消失

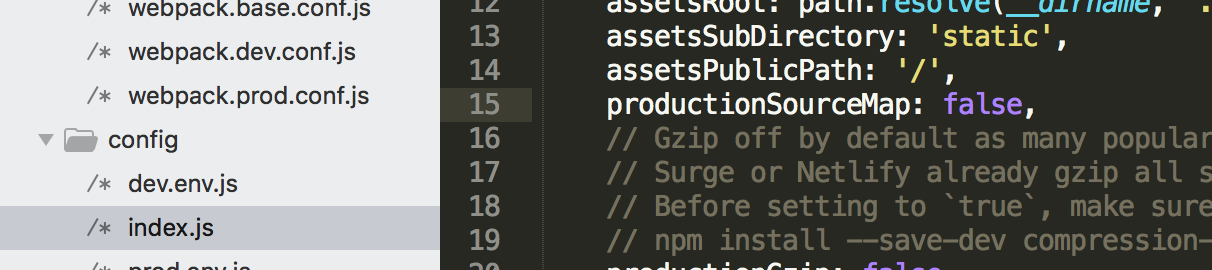


39.获取元素高度

<div ref=”XXX”></div>

mounted获取不到可以在updated获取  
 this.$refs.XXX. clientHeight  
 修改的话要加’px’



40.打包后source.map过大  
 <https://segmentfault.com/q/1010000008082747>  
 

41.子组件传空字符串

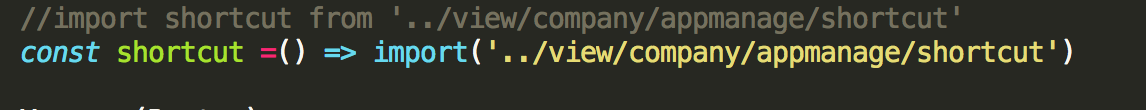
要双引号里包裹单引号,反过来也行  
 

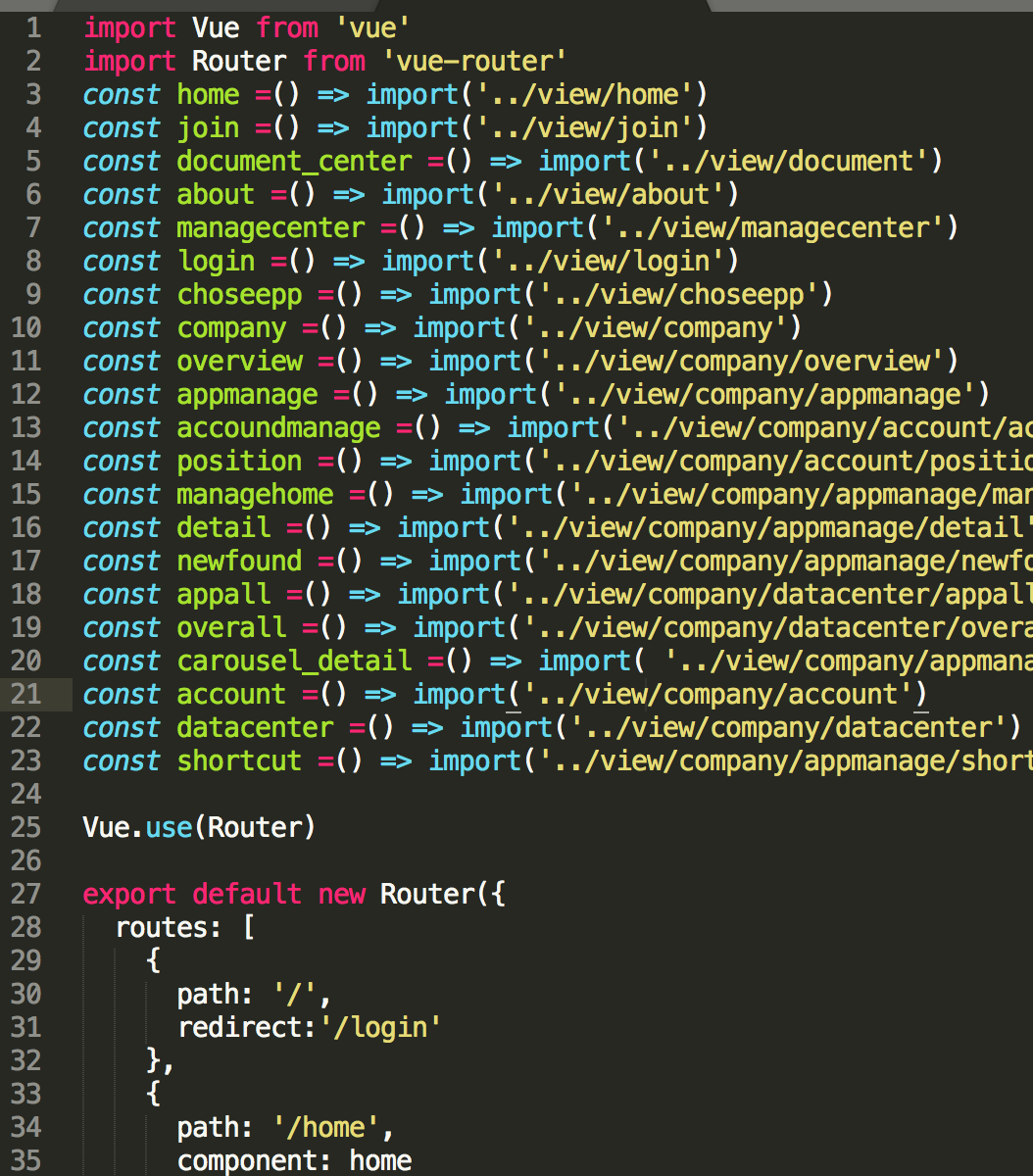
42.当一个页面既有http也有https,用a标签跳进来浏览器会报错,用路由组件导航

43.路由懒加载(异步路由组件)

网站登录页面不要用异步路由,会清除掉保存的信息localstorage

https://router.vuejs.org/zh-cn/advanced/lazy-loading.html

第一步: npm install --save-dev babel-plugin-syntax-dynamic-import  
 第二步: 路由的index文件:由index from改成const from 打包后体积会减小  
 



44.异步组件 (减小打包体积)

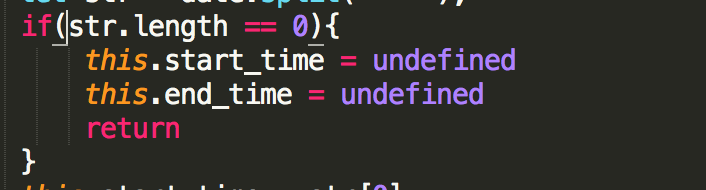
http://blog.csdn.net/qq20004604/article/details/56496597



es6写法:

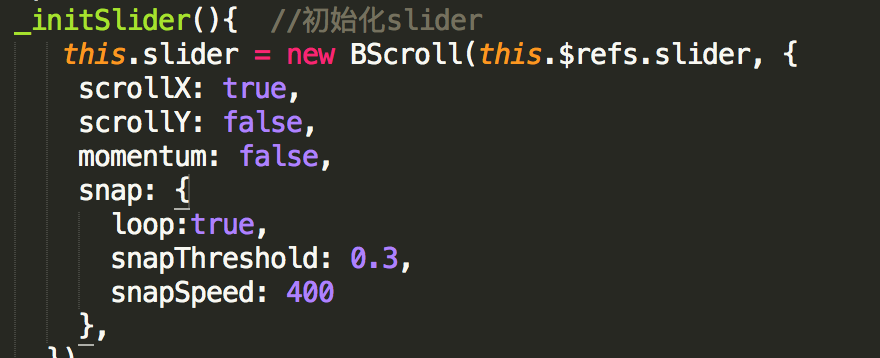


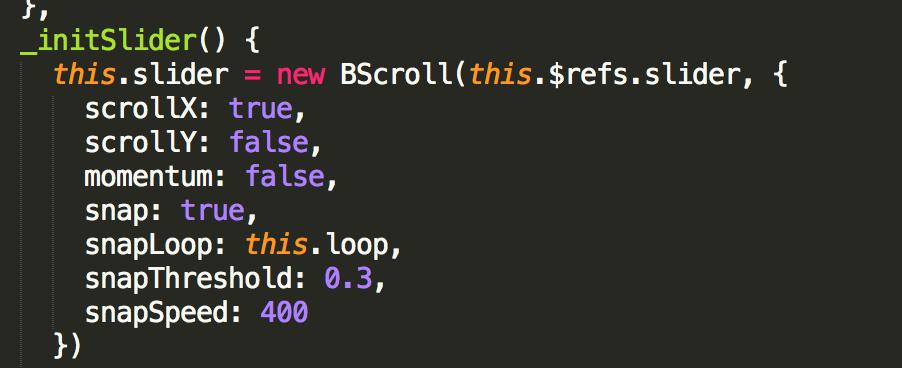
45.ajax变量为空不传递参数  
 wantch这个变量 如果为空则赋值为undefined,



46.滚动插件better-scroll  
 1.X版本设置无限轮播:在snap中设置

<https://github.com/ustbhuangyi/better-scroll/blob/master/example/components/slide/slide.vue>



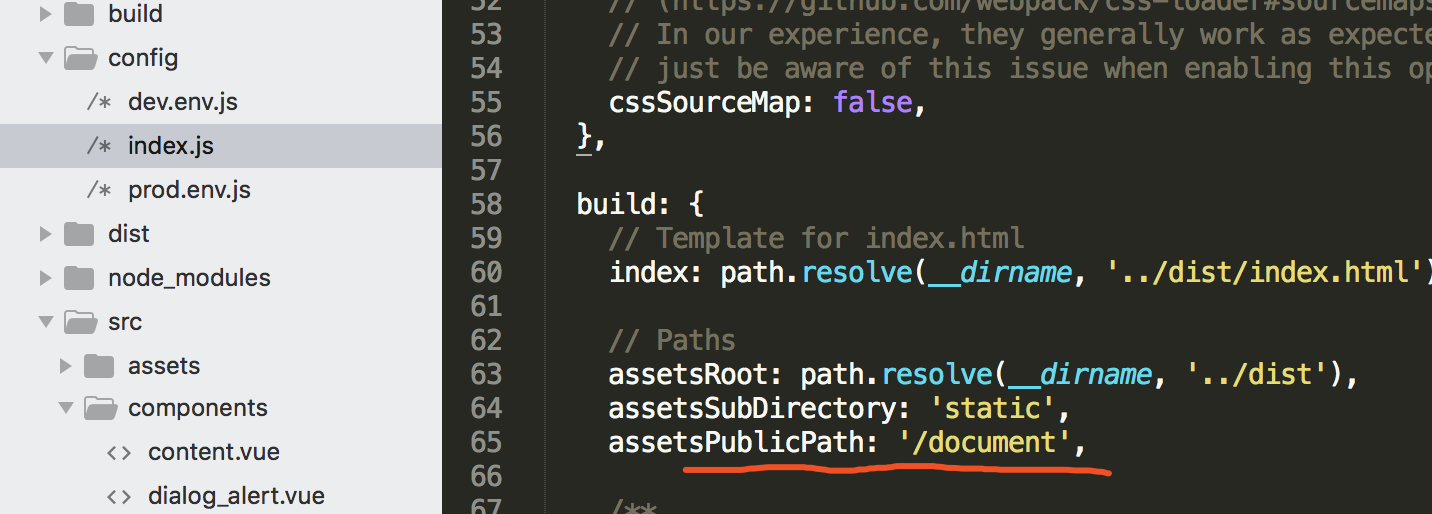
1. X版本设置轮播方法  
    

47.打包上线有缓存  
 服务端配置:不保留index.html的缓存

48.打包文件挂在到服务器子目录下

需要把图片放在assets目录下

1.assetsPublicPath改成子目录名



2.nginx 服务器新增一个location root仅指向根目录不指向子目录,try\_files包含子目录

location /document {

root /home/zxdeng;

# rewrite ^/document/?$ /document;

try\_files $uri $uri/ /document/index.html;

index index.html;

}

参考: <http://wudong158.cn.segmentfault.com/q/1010000011901432>

第二种方法:

assetsPublicPath设置为’./’

router设置base为所在目录

<http://www.jb51.net/article/126861.htm>

<https://router.vuejs.org/zh-cn/api/options.html#mode>

49.重新初始化data

Object.assign(this.$data, this.$options.data())

50.在新窗口打开Vue页面

window.open(window.location. origin + '/a/b/c')

例子: window.open(window.location. origin + '/#/company/overview')

把变量存在localStorage中另一个页面就能用了

https://segmentfault.com/q/1010000010116819/