https://segmentfault.com/a/1190000012249890

[vue-cli打包后的思索--代码优化](https://segmentfault.com/a/1190000012249890)

* [vue-cli](https://segmentfault.com/t/vue-cli/blogs)

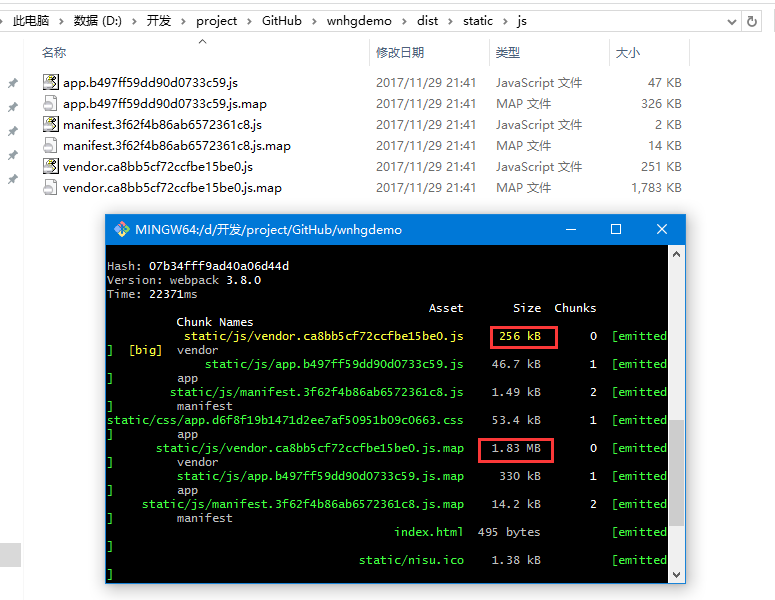
* [vue.js](https://segmentfault.com/t/vue.js/blogs)

[**冰扬**](https://segmentfault.com/u/bingyang519) 2017年12月01日发布

* 赞  |   0收藏  |  18
* 577 次浏览

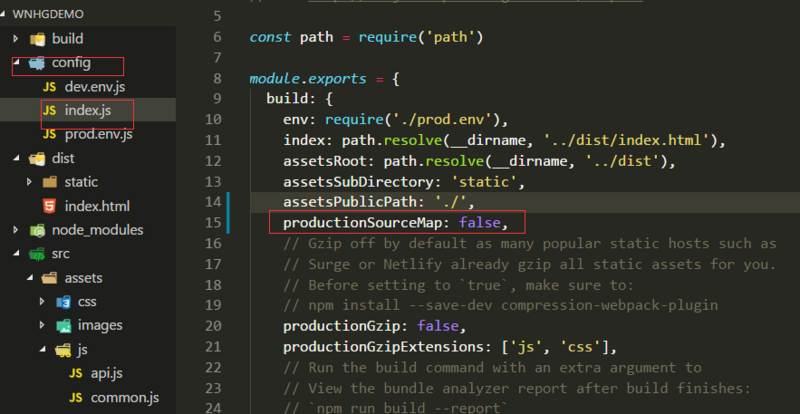
书接上文，[利用http-server测试vue-cli打包后的项目](https://segmentfault.com/a/1190000012249873)。已经可以用http-server测试打包后的项目了，可是看到打包后的代码。。。

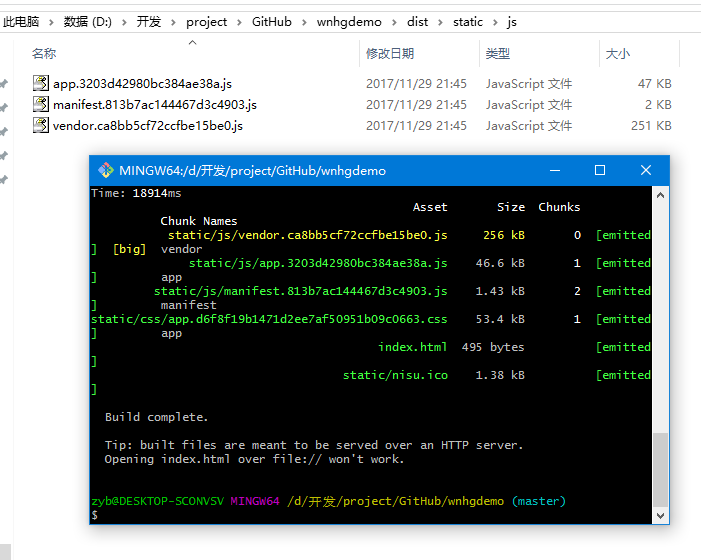
.map文件

* 第一次打包，发现打包后的文件怎么这么大，.map文件又是什么鬼？  
     
  
* 经过一顿搜索，发现[js的map文件](https://www.cnblogs.com/weloveshare/p/5753717.html" \t "_blank).原来map文件只是帮助我们调试用的，毕竟打包后的代码都是压缩的，调试起来很不方便。测试没问题，正式上线时，我们完全可以去掉这个文件。



* 怎么去掉这个文件呢？一个一个删？no！no！  
     
  只需把config/index.js中的productionSourceMap: true改为productionSourceMap: false即可



* 重新打包的代码  
     
  

代码太大

打包后的代码，体积还是有点大？如何减小体积，提高首屏加载速度呢

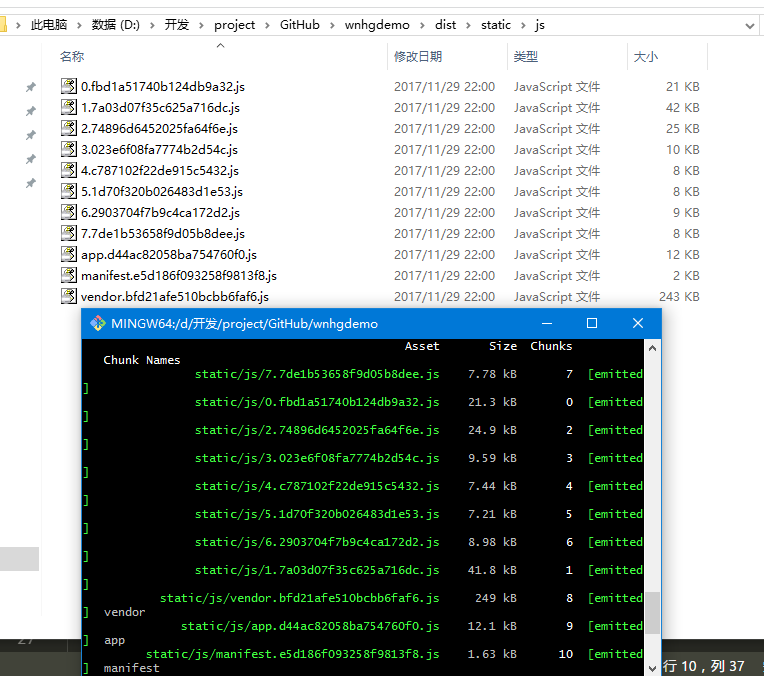
* 路由懒加载

之前没使用路由懒加载功能，导致所有的代码都打包在了一个文件中，又查看了一下vue-router的[路由懒加载](https://router.vuejs.org/zh-cn/advanced/lazy-loading.html" \t "_blank),参照其中的示例，把路由组件的引入方式作了一些更改。

* + 更改前  
    
  + 更改后

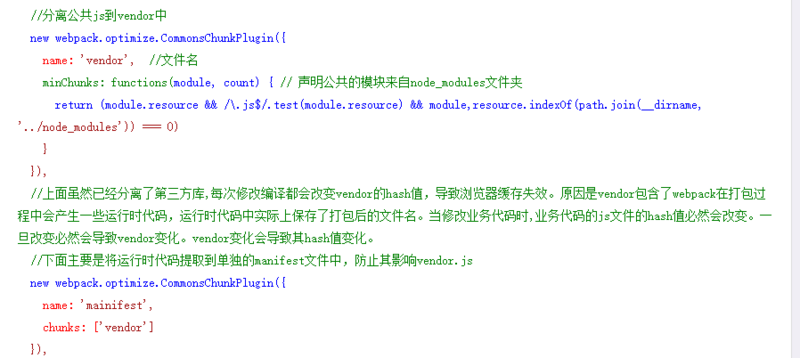


* + 打包后

  
使用路由懒加载打包的代码，数量上虽然多了，但是通过查看引入情况，首次加载，并不会全部加载，每次打开一个新路由，才会加载相应的js,实现了按需加载，提高了渲染效率。

vendor.js

即使使用了路由懒加载，发现其中vendor这个js文件貌似一直都没变化，还是特别大，这个又是什么文件呢？经过一轮搜索后才发现其真面目。  
　　参考：[vue-cli的webpack模板项目配置文件分析](http://blog.csdn.net/hongchh/article/details/55113751)、[vue-cli中的webpack配置](https://segmentfault.com/a/1190000008779053)  
原来这个文件是我们npm安装的那些库



如何减小这个文件的体积呢？我还没研究透，大家还是移步这个问题，自己去探索吧。