当前我们company情况

1、需要提取公共模块common.js

2、提取公共模块cssreset.css

3、css采用scss编译

[**命令行生成vue2.0 -cli项目框架**](http://blog.csdn.net/zzwwjjdj1/article/details/52687488)

1. npm install -g vue-cli
2. vue init webpack vue2.0(默认安装2.0，安装这个)

vue init webpack#1.0 vue2.0(安装1.0)

1. npm install(安装依赖的东西)
2. npm run dev运行开发服务器环境
3. npm run build 生成生产环境

提醒

1、1个cli项目只有一个index.html,App.vue,main.js

2、所有组件放在：index.html中div#app里面

**文档：**

2.0官网文档：<https://vuejs.org/guide/>

2.0中文文档：<https://vuefe.cn/guide/>

**定义与使用组件步骤：**

1、定义组件，格式包括<template>,<script>,<style>

2、引入组件 import

3、注册组件 components

4、使用组件 <firstcomponent></firstcomponent>

app.vue（使用FirstComponent组件）

<template>  
 <div id="app">  
 <img src="./assets/logo.png">  
 <FirstComponent></FirstComponent>  
 </div>  
</template>  
<script>  
**import** FirstComponent **from** './components/FirstComponent'  
**export default** {  
 name: 'app',  
 components: {  
 FirstComponent  
 }  
}

FirstComponent.vue（自己定义的）

<template>  
<h2>{{text}}</h2>  
</template>  
<script>  
 **export default**{  
 data () {  
 **return** {  
 text: '这是我的第一个组件'  
 }  
 }  
 }  
</script>  
<style scoped>  
 h2{  
 background: red;  
 }  
</style>

**项目使用SASS语法**

1. npm install node-sass –save-dev
2. npm install sass-loader –save-dev
3. webpack配置

{  
 test: /\.scss$/,  
 loader: "style!css!sass"  
},

1. 使用

<style lang='scss'>  
@import "./assets/scss/test.scss";//引入外部scss文件  
#app1 {  
 font-family: 'Avenir', Helvetica, Arial, sans-serif;  
 -webkit-font-smoothing: antialiased;  
 -moz-osx-font-smoothing: grayscale;  
 text-align: center;  
 color: #2c3e50;  
 margin-top: 60px;  
 & .red{  
 color:red;  
 }  
}  
</style>

1. autoprefixer的添加

webpack配置

vue: {  
 loaders: utils.cssLoaders(),  
 postcss: [  
 require('autoprefixer')({  
 browsers: ['last 2 versions','Android >= 4.0','iOS >= 7']  
 })  
 ]  
}

这个效果有用（下面代码，用的是css而不是scss）

<style scoped>  
 h2{  
 background: red;  
 display: flex;  
 }  
</style>

生成

<style type="text/css">  
 h2[data-v-34c8e960]{  
 background: red;  
 display: -webkit-box;  
 display: -webkit-flex;  
 display: -ms-flexbox;  
 display: flex;  
 }  
</style>

但是下面这个却没有效果（部分没有比如display）

<style lang='scss'>  
@import "./assets/scss/test.scss";  
#app1 {  
 font-family: 'Avenir', Helvetica, Arial, sans-serif;  
 text-align: center;  
 color: #2c3e50;  
 margin-top: 60px;  
 diplay:flex;  
 animation-name: dropAnim1;  
 & .red{  
 color:red;  
 }  
}  
</style>

生成效果

<style type="text/css">  
 html, body {  
 background: red;  
 -webkit-animation-name: dropAnim1;  
 animation-name: dropAnim1;  
 diplay: flex;  
 }  
 #app1 {  
 font-family: 'Avenir', Helvetica, Arial, sans-serif;  
 text-align: center;  
 color: #2c3e50;  
 margin-top: 60px;  
 diplay: flex;  
 }  
 #app1 .red {  
 color: red;  
 -webkit-animation-name: dropAnim1;  
 animation-name: dropAnim1;  
 }  
</style>

解决连接：

<https://github.com/vuejs/vue-loader/issues/227>

<https://github.com/vuejs/vue-loader/issues/234>

<https://github.com/vuejs/vue-loader/issues/424>

[webpack如何提取vue组件的css到独立文件中](https://segmentfault.com/q/1010000005363929)

**路由vue-router**

1、安装

npm install vue-router --save-dev

2、使用

在main.js中引入

**import** Vue **from** 'vue'  
**import** App **from** './App'  
**import** VueRouter **from** 'vue-router'  
Vue.use(VueRouter)  
// 定义组件, 也可以像教程之前教的方法从别的文件引入  
**const** First = { template: '<div><h2>我是第 1 个子页面</h2></div>' }  
**const** Second = { template: '<div><h2>我是第 2 个子页面</h2></div>' }  
// 创建一个路由器实例  
// 并且配置路由规则  
**const** router = **new** VueRouter({  
 mode: 'history',  
 base: \_\_dirname,  
 routes: [  
 {  
 path: '/first',  
 component: First  
 },  
 {  
 path: '/second',  
 component: Second  
 }  
 ]  
})  
// 路由器会创建一个 App 实例，并且挂载到选择符 #app 匹配的元素上。  
**new** Vue({  
 router: router,  
 render: h => h(App)  
}).$mount('#app')

App.vue

<template>  
 <div id="app">  
 <img src="./assets/logo.png">  
 <FirstComponent></FirstComponent>  
 <ul>  
 <li><router-link to="/first ">点我跳转到第一页</router-link></li>  
 <li><router-link to="/second">点我跳转到第二页</router-link></li>  
 </ul>  
 <router-view></router-view>  
 </div>  
</template>

**参考资料：**

大致了解vue.js -cli

<http://www.larabase.com/collection/3/post/128>

[Vue2.0 新手完全填坑攻略——从环境搭建到发布](http://www.jianshu.com/p/5ba253651c3b)

[Vue.js 2.0 基本入门](http://www.thatyou.cn/vue-js-2-0-%E5%9F%BA%E6%9C%AC%E5%85%A5%E9%97%A8/)

[国内有哪些公司在用Vue.js，有什么心得?](https://www.zhihu.com/question/38213423)

[Vue 组件化开发实践](http://gold.xitu.io/entry/55f77eb460b28e6a6f0f4f86)

[Vue + webpack 项目实践](http://jiongks.name/blog/just-vue/)

[webpack入坑之旅（六）配合vue-router实现SPA](http://www.tuicool.com/articles/yI322iq)

[vue-router 快速入门](http://www.cnblogs.com/axl234/p/5899952.html)

IDE工具

<http://www.jianshu.com/p/142dae4f8b51>

e6相关

[VUE.JS组件化开发实践](http://www.cnblogs.com/rock-roll/p/5692891.html)

[Laravel 5.1 中使用 Vue2.0 组件化开发（配置）](http://www.open-open.com/lib/view/open1471589613718.html)

[vue.js2.0路由文档](http://router.vuejs.org/)

[Vue.js 开发实践：实现精巧的无限加载与分页功能](https://segmentfault.com/a/1190000005351971)

[用vue+webpack+es6+sass实现Cnode网站(一)](https://segmentfault.com/a/1190000006000118)

[在VUE2.0中使用PostCSS](http://gold.xitu.io/post/581bfc368ac247004fe174af)

github源码参考

<https://github.com/825618507/vue-app>（入门适合）

<https://github.com/zerqu/qingcheng>

[webpack+vue.js重构的cnode社区的单页面(spa)应用](https://github.com/cwsjoker/Cnode-vue-spa)

[知乎日报](https://github.com/luchanan/vue-zhihu-daily)

<https://github.com/bluefox1688/vue-cli-multi-page>