1. vue-cli创建项目

npm i -g vue-cli

vue init webpack projectname

1. 按需引入mint-ui组件，完成头部区域的制作。

npm install babel-plugin-component -D

import { Header } from 'mint-ui'

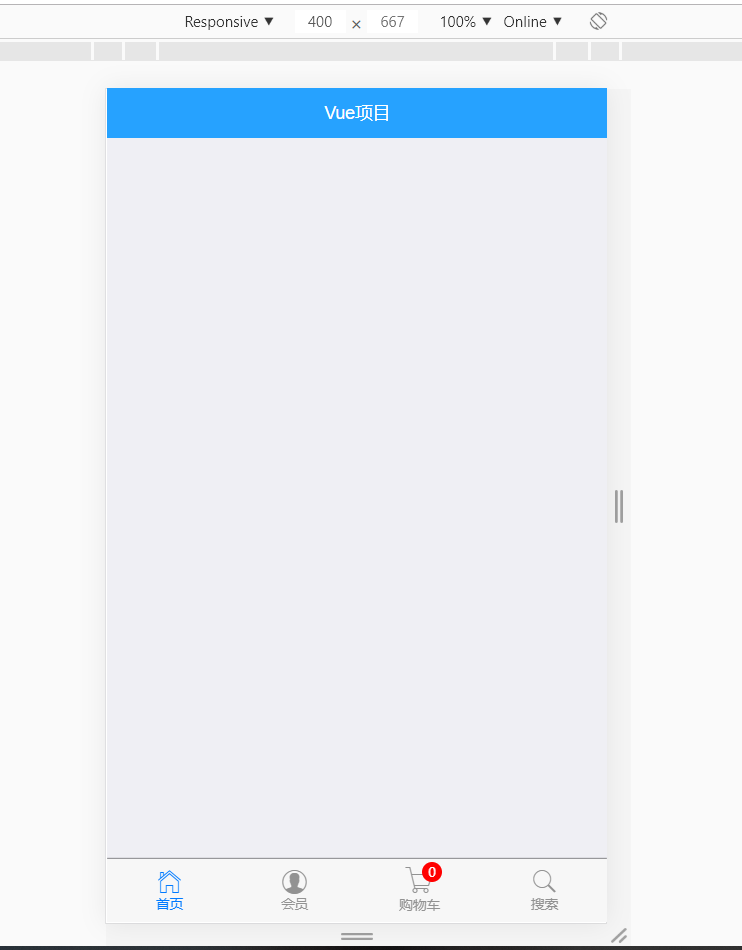
import 'mint-ui/lib/style.css'

Vue.component(Header.name, Header)

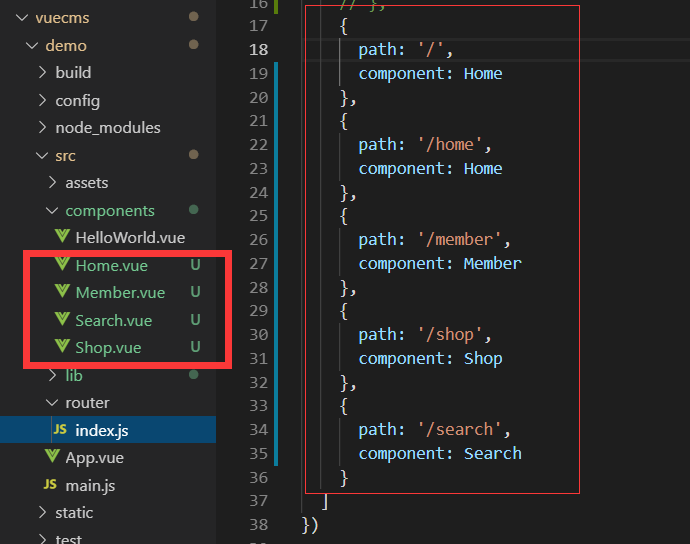
1. 下载mui引入，完成底部区域的制作。

import './lib/mui/css/mui.min.css'

import './lib/mui/css/icons-extra.css'



1. 制作中间部分，主要就是创建vue组件，规定好路由。



1. 跨域问题解决

引入axois，访问接口

import axios from 'axios'

Vue.prototype.$axios= axios

（1）config下的index.js中添加

proxyTable: {

'/api': {

target:'http://phps.php.com/tp5/public/index.php',

// secure: false, // 如果是https接口，需要配置这个参数

changeOrigin:true,

pathRewrite:{

'^/api': ''

}

}},

（2）访问后端接口，url必须加上‘api’

this.$axios({

methods: 'get',

url: 'api/index/index/index',

}).then(res => {

console.log(res);

})

1. 路由传参

（1）路由设置：

{

path: '/home/listinfo/:id',

component: ListInfo,

}

（2）参数传递:

<li v-for="item in list" :key="item.id">

       <router-link :to="'/home/listinfo/' + item.id">

        </router-link>

 </li>

（3）接收参数：

data() {

return {

id: this.$route.param.id

}

}

1. 点击加载更多时，使用tp5中page获得每一页信息，在前端data中定义page记录获得的数据到第几页，然后再次调用获得数据的接口，把得到的数据用数组拼接显示。

<mt-button type="danger" size="large" plain @click="getMore">加载更多...</mt-button>

this.content = this.content.concat(res.data);

getMore() {

this.page ++;

this.getPing();

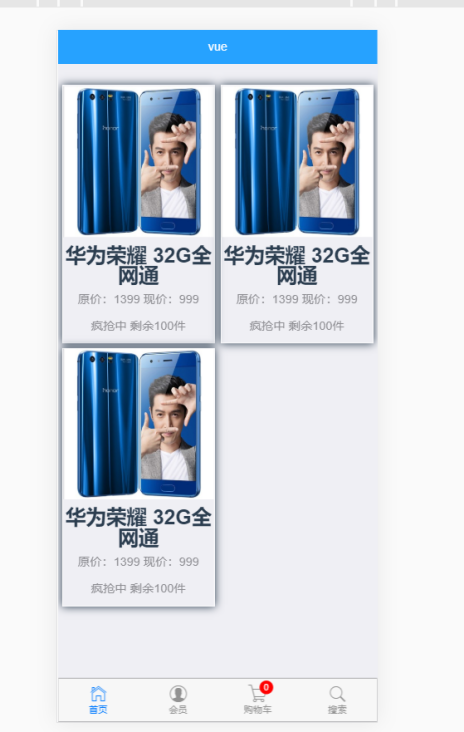
}

1. 父组件调用子组件的方法并传参

<ImgList ref="mychild"></ImgList>

this.$refs.mychild.getImgList(num);

1. 实现商品的两列布局



Html代码

<div class="div1">

<div class="shopitem">

<img src="http://phps.php.com/tp5/vendor/image/phone1.jpg">

</div>

</div>

样式

.div1{

margin-bottom: 50px;

display: flex;

/\*  元素在必要的时候拆行或拆列。 \*/

flex-wrap: wrap;

/\* 各个元素之间留空白 \*/

justify-content: space-between;

padding: 5px;

}

.shopitem{

/\* 宽度设置为49%，让两个商品在一行 \*/

width: 49%;

/\* 加边框和阴影 \*/

border: 1px solid #fff;

box-shadow: 0 0 10px;

margin-bottom: 5px;

}

img{

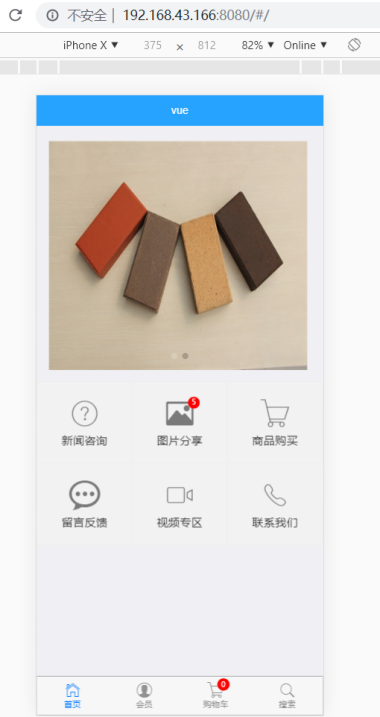
width: 100%;

}

1. 在手机中调试
2. 保证手机和开发电脑处于同一个WIFI（手机热点也可）
3. 在package.json文件中的“dev”添加一个--host指令，设置为电脑的ip地址（ipconfig）

"dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js --host 192.168.43.166",

1. 运行代码，[http://192.168.43.166:8080/#/](http://192.168.43.166:8080/" \l "/)，把此地址发于手机点击运行测试。



1. 跳转
2. 使用<router-link>

<router-link class="shopitem" v-for="(item, id) in list"

:key="id" :to="'/home/shopinfo/' + item.id">

</router-link>

1. 使用this.$router.push()四种方式

<div class="shopitem" v-for="(item, id) in list" :key="id"

@click="getInfo(item.id)">

</div>

getInfo(id) {

前三种路由形式/home/shopinfo/4

1. this.$router.push("/home/shopinfo/" + id);
2. this.$router.push({path: "/home/shopinfo/" + id});

（3）this.$router.push({

name: 'shopinfo', //name值不是路由名称，是定义路由时的name

params: {'id': id}

})

}

（4）注意第四这种特殊，跳转的路由？后加参数,/home/shopinfo?id=4

this.$router.push({

path: "/home/shopinfo",

query: {'id': id}

})

注意：this.$route是参数对象，如params，query

this.$router主要用于跳转到新的路由

1. 使用vuex
2. 添加store文件夹，并设置index.js文件

import Vue from 'vue'

import Vuex from 'vuex'

Vue.use(Vuex)

export default new Vuex.Store({

state: {

test: '测试一下',

},

mutations: {

}

})

1. 在mian.js中注册store

import store from './store'

new Vue({

el: '#app',

store,

})

1. 访问store中的数据

<div>{{$store.state.test}}</div>

1. 打包

npm run build

注意：使用了mui，打包会出现错误，找到mui的css文件下面的mui.css和mui.min.css文件，把里面background-image中的单引号改为双引号。