# 《vue》说明文档

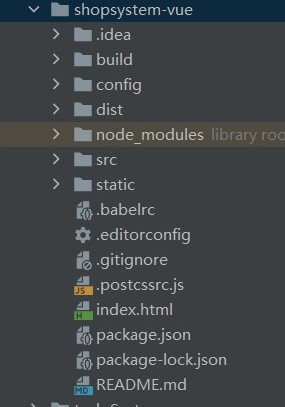
### 1、学号：

### 2、姓名：

* 技术框架 Vue+VantUi +Mock

前端部分使用了当下流行的Vue框架，而UI框架部分使用了移动端的VantUI，能实现自适应布局，非常符合当下人们的需求。

目录架构：



* 界面功能

登录与注册界面

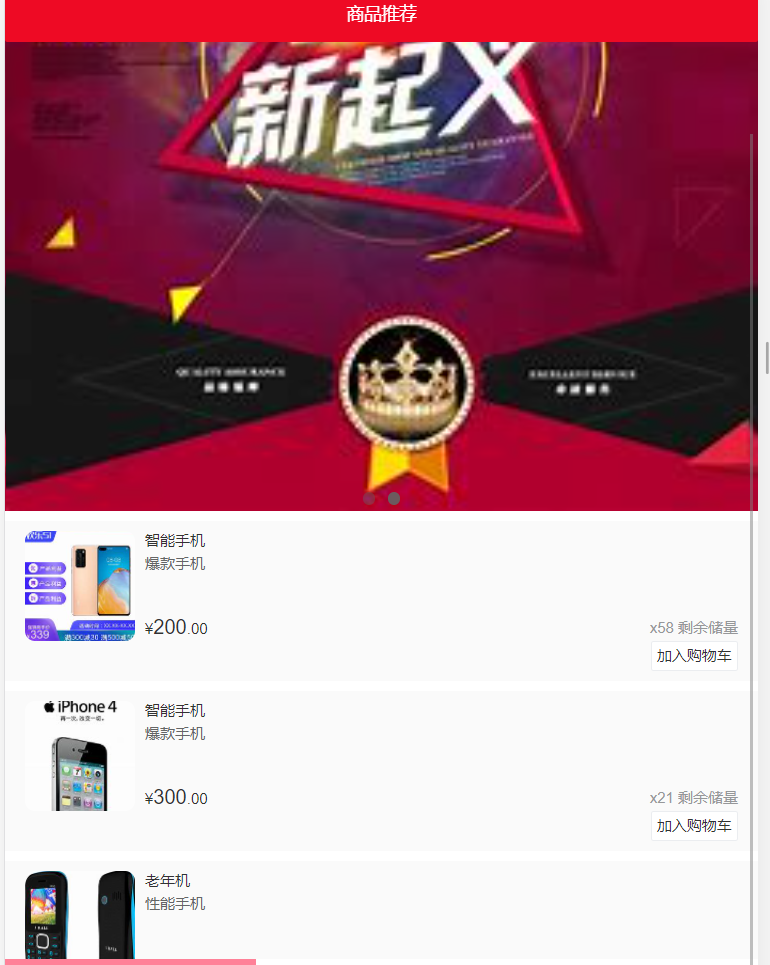




关键代码

<template>  
<div class="login">  
 <van-tabs type="card">  
 <van-tab title="DENGLU❤">  
 <van-form @submit="onSave">  
 <van-field  
 v-bind:value="userAccount"  
 name="user\_account"  
 label="用户名"  
 placeholder="用户名"  
 :rules="[{ required: true, message: '请填写用户名' }]"  
 />  
 <van-field  
 v-model="password"  
 type="password"  
 name="password"  
 label="密码"  
 placeholder="密码"  
 :rules="[{ required: true, message: '请填写密码' }]"  
 />  
 <div style="margin: 16px;">  
 <van-button round block type="info" native-type="submit">提交</van-button>  
 </div>  
 </van-form>  
 </van-tab>  
 <van-tab title="ZHUCE😀">  
 <van-form @submit="onSubmit">  
 <van-field  
 v-model="username"  
 name="username"  
 label="用户名"  
 placeholder="用户名"  
 :rules="[{ required: true, message: '请填写用户名' }]"  
 />  
 <van-field  
 v-model="userAccount"  
 name="userAccount"  
 label="账号名"  
 placeholder="账号名"  
 :rules="[{ required: true, message: '请填写账号' }]"  
 />  
 <van-field  
 v-model="telephone"  
 type="text"  
 name="phone"  
 label="手机号"  
 placeholder="手机号"  
 :rules="[{ required: true, message: '请填写手机号' }]"  
 />  
 <van-field  
 v-model="password"  
 type="password"  
 name="password"  
 label="密码"  
 placeholder="密码"  
 :rules="[{ required: true, message: '请填写密码' }]"  
 />  
 <van-field  
 v-model="confirmPass"  
 type="password"  
 name="confirmPassword"  
 label="确认密码"  
 placeholder="再输入一次你的密码哦"  
 :rules="[{ required: true, message: '请确认',validator:Validator }]"  
 />  
 <van-field  
 v-model="address"  
 type="text"  
 name="address"  
 label="输入地址"  
 placeholder="请输入你的地址"  
 :rules="[{ required: true, message: '请输入你的地址',validator:Validator }]"  
 />  
  
 <div style="margin: 16px;">  
 <van-button round block type="info" native-type="submit">提交</van-button>  
 </div>  
  
  
 </van-form>  
 </van-tab>  
 </van-tabs>  
 </div>  
  
  
</template>  
  
<script>  
 import {areaList} from "@vant/area-data";  
  
 export default {  
 name: 'login',  
 data() {  
 return {  
 areaList: areaList,  
 searchResult: [],  
 username: '',  
 password: '',  
 confirmPass: '',  
 telephone: '',  
 address: "",  
 userAccount: ''  
 };  
 },  
 methods: {  
 async onSave(e) {  
 let params = {  
 user\_account: e.user\_account,  
 password: e.password}  
 console.log(params)  
  
 const res = await this.$http.postLogin(params);  
 console.log(res);  
 if (res.ret === 0) {  
 localStorage.setItem('token', res.token);  
 }  
 await this.$router.push({path:"/main"});  
  
 console.log(e);  
 },  
 async onSubmit(e) {  
 if (!e.address) {  
  
 }  
 let user = {  
 username: e.username,  
 user\_account: e.userAccount,  
 password: e.password,  
 phone: e.phone,  
 receiveAddress: e.address  
 }  
 await this.$http.postRegister(user);  
 // await this.$http.getAllProducts();  
 // console.log('submit', values);  
 },  
 areaclick() {  
 console.log(1)  
 },  
 areaInfo(values) {  
 console.log(values);  
 },  
 Validator() {  
 return this.password === this.confirmPass;  
 },  
 onChangeDetail(val) {  
 if (val) {  
 this.searchResult = [  
 {  
 },  
 ];  
 } else {  
 this.searchResult = [];  
 }  
 },  
 },  
 };  
</script>  
  
<style scoped>  
  
</style>

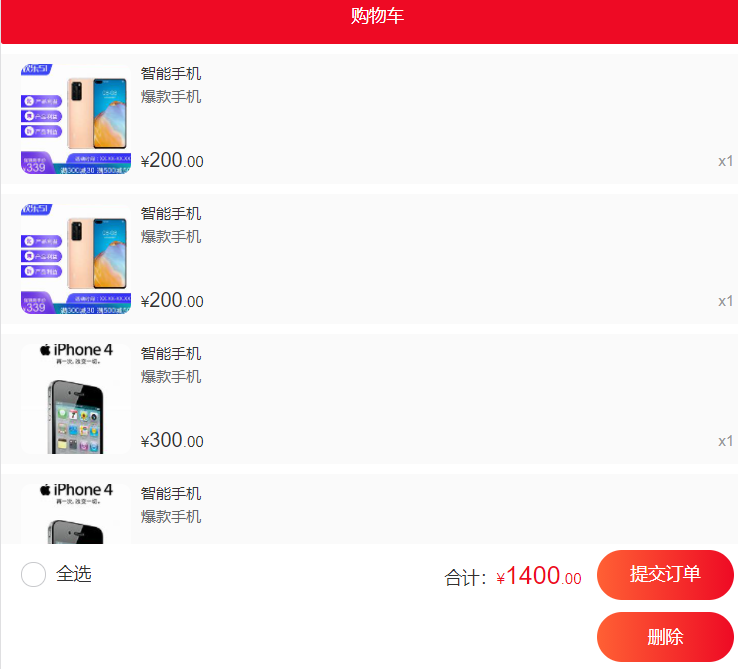
商品页面



关键代码

<template>  
<div>  
 <van-sticky>  
 <van-button type="danger" style="width: 100%">商品推荐</van-button>  
 </van-sticky>  
 <swiper>  
 <swiper-item v-for="(item, index) in banners" :key="index">  
 <img :src="item.img" alt=""/>  
 </swiper-item>  
 </swiper>  
  
 <van-card  
 :num="item.stock + ' 剩余储量'"  
 :price="item.price"  
 :desc="item.product\_desc"  
 :title="item.product\_name"  
 :thumb="item.image"  
 v-for="(item, index) in goods"  
 :key="index"  
 >  
 <template #tags>  
 <van-tag  
 plain  
 type="danger"  
 v-for="(item, index) in item.keyWord"  
 :key="index"  
 >{{ item }}</van-tag  
 >  
</template>  
  
<template #footer>  
<van-button size="mini" @click="intoCart(item.product\_id, item.price,index)"  
 >加入购物车</van-button  
 >  
</template>  
</van-card>  
</div>  
</template>  
<script>  
 import { Swiper, SwiperItem } from "./Swiper/useSwiper";  
 import { get } from "../utils/http"  
 import {Notify} from "vant";  
  
 export default {  
 components: {  
 Swiper,  
 SwiperItem,  
 },  
 data() {  
 return {  
 active: 0,  
 banners: [{img: 'https://tse2-mm.cn.bing.net/th/id/OIP.wzzaurqikEdc67s1MQ0wegHaGo?w=188&h=180&c=7&o=5&dpr=1.25&pid=1.7'},{img: 'https://tse2-mm.cn.bing.net/th/id/OIP.wzzaurqikEdc67s1MQ0wegHaGo?w=188&h=180&c=7&o=5&dpr=1.25&pid=1.7'}],  
 goods: [],  
 };  
 },  
 async created() {  
 let allProducts = await this.$http.getAllProducts();  
  
 this.goods = allProducts.data;  
 console.log(this.goods)  
 },  
 props: {  
 list: {  
 default() {  
 return [  
 ];  
 },  
 },  
 },  
 methods: {  
 async intoCart(id, price,index) {  
 let res = await this.$http.addToMyCart({amount: 1, productId: id, totalPrice: 1 \* price, price: price});  
 if (res.ret === 0) {  
 Notify({type: 'success', message:'加入购物车成功'});  
 }  
 this.goods[index].stock -= 1;  
  
 },  
 },  
 };  
</script>  
<style scoped>  
 .todo-container li {  
 border: solid 1px white;  
 background-color: beige;  
 margin-top: 1em;  
 padding: 0.2em;  
 }  
 .main {  
 font-size: 0.7em;  
 }  
</style>

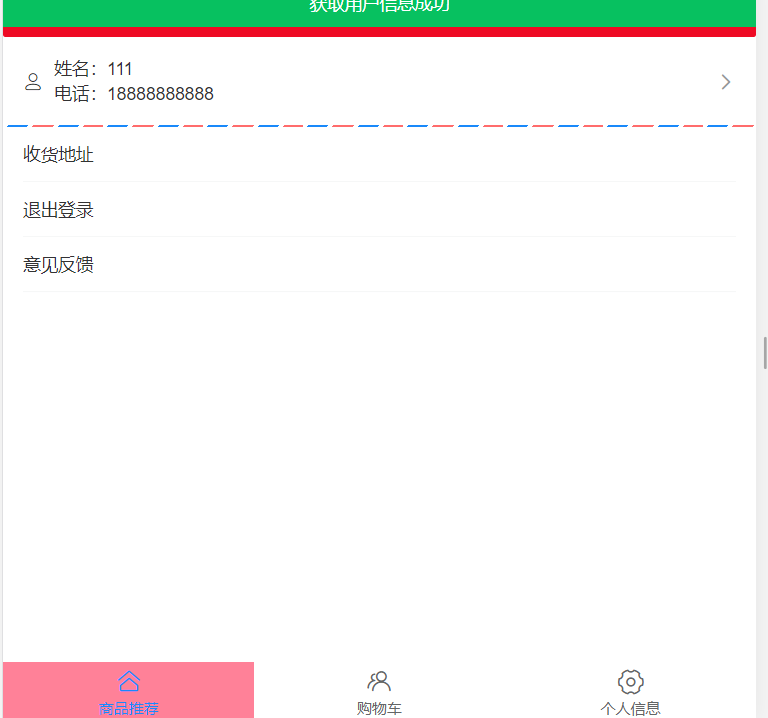
购物车界面



关键代码

<template>  
<div class="cart">  
 <van-sticky>  
 <van-button type="danger" style="width: 100%"> 购物车 </van-button>  
 </van-sticky>  
  
 <van-card  
 :num="item.amount"  
 :price="item.price"  
 :desc="item.product\_desc"  
 :title="item.product\_name"  
 :thumb="item.image"  
 v-for="(item, index) in cartGoods"  
 :key="index"  
 >  
 <template #tags>  
 <van-tag  
 plain  
 type="danger"  
 v-for="(item, index) in item.keyWord"  
 :key="index"  
 >{{ item }}</van-tag  
 >  
</template>  
  
</van-card>  
<van-submit-bar  
 :price="totalPrice"  
 button-text="提交订单"  
 @submit="onSubmit"  
 class="van-submit-bar"  
 >  
  
 <van-checkbox v-model="checked">全选</van-checkbox>  
</van-submit-bar>  
  
<van-submit-bar  
 button-text="删除"  
 @submit="onDelete"  
 class="van-submit-bar1"  
 >  
  
</van-submit-bar>  
  
  
</div>  
</template>  
<script>  
 import { get } from "../utils/http";  
 export default {  
 props: {},  
 data() {  
 return {  
 active: 0,  
 show: false,  
 value: null,  
 checked: false,  
 cartGoods: [],  
 totalPrice: 0  
 };  
 },  
 methods: {  
 onSubmit() {},  
 async onDelete() {  
 if (this.checked) {  
 let size = this.cartGoods.length;  
 let cartCopy = this.cartGoods;  
 this.cartGoods = [];  
 this.totalPrice = 0;  
 for (let i = 0; i < size; i++) {  
 await this.$http.deleteMyCart({cartId:cartCopy[i].cart\_id});  
 }  
 }  
 }  
 },  
 async created() {  
 // let cartArr = JSON.parse(sessionStorage.getItem("cartGoods")) || [];  
 let cartArr = await this.$http.getMyCart();  
 this.cartGoods = cartArr.data;  
 console.log(this.cartGoods.length)  
 for (let i = 0; i < this.cartGoods.length; i++) {  
 this.totalPrice += ~~this.cartGoods[i].price \* 100;  
 }  
 },  
 };  
  
</script>  
<style scoped>  
 .van-submit-bar {  
 bottom: 100px;  
  
 }  
 .van-submit-bar1 {  
 bottom: 50px;  
 }  
 .cart {  
 padding-bottom: 100px;  
 }  
</style>

我的界面



关键代码

<template>  
<div>  
 <van-sticky>  
 <van-button type="danger" style="width: 100%">我的信息</van-button>  
 </van-sticky>  
  
 <van-contact-card  
 type="edit"  
 :name="user.username"  
 :tel="user.phone"  
 @click="showMore"  
 />  
 <van-contact-list  
 v-model="chosenContactId"  
 :list="list"  
 default-tag-text="默认"  
 v-if="isEdit"  
 @Add="onAdd"  
 class="van-contact-list"  
 />  
  
 <van-list :finished="finished">  
 <van-cell v-for="(item,index) in listName" :key="item" @click="test(index)" :title="item"/>  
 </van-list>  
  
  
 </div>  
</template>  
<script>  
 import {Notify} from "vant";  
  
 export default {  
 props: {},  
 data() {  
 return {  
 isEdit: false,  
 active: 0,  
 loading: false,  
 finished: false,  
 listName: ["收货地址", "退出登录", "意见反馈"],  
 chosenContactId: "2",  
 user: {},  
 list: [  
  
 ],  
 };  
 },  
 computed: {  
 showWho() {  
 return {  
 name: "张三",  
 tel: "13000000000",  
 };  
 },  
 },  
 methods: {  
 showMore() {  
 this.isEdit = !this.isEdit;  
 },  
 async logout() {  
 console.log('我在这');  
 if (localStorage.getItem('token')) {  
 localStorage.removeItem('token');  
 const res = await this.$http.deleteLogout();  
 console.log(res);  
 await this.$router.push('/');  
 }  
 },  
 test(t) {  
 switch (t) {  
 case 0:  
 Notify({type: 'success', message: '点击上方'})  
 break;  
 case 1:  
 this.logout();  
 break;  
 case 2:  
 Notify({type: 'warning', message: '暂未开通该功能'})  
 break;  
 default:  
 break;  
 }  
 console.log(t)  
 },  
 onAdd() {  
  
 }  
  
 },  
 async created() {  
 let userInfo = await this.$http.getUserinfo();  
 this.user = userInfo.data[0];  
 const res = await this.$http.getMyAddress();  
 const user = await this.$http.getUserinfo();  
 console.log(user)  
 let i = res.data.length;  
 for (let j = 1; j<= i; j++) {  
 console.log(res.data)  
 if (j == 1) {  
 this.list.push({id: j - 1, name: user.data[0].username, tel: res.data[j - 1].address, isDefault: true})  
 } else {  
 this.list.push({id: j - 1, name: user.data[0].username, tel: res.data[j - 1].address})  
 }  
 }  
  
 }  
 };  
</script>  
<style scoped>  
 .todo-container li {  
 border: solid 1px;  
 background-color: beige;  
 margin-top: 1em;  
 border-radius: 2em;  
 padding: 1em;  
 }  
  
 .van-contact-list >>> .van-contact-list\_\_bottom {  
 display: none;  
 }  
  
 .van-contact-list {  
 padding-bottom: 20px;  
 }  
</style>

Mock使用：

// mock.js

// 引入mockjs

const Mock = require('mockjs');

const login = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const register = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const addAddress = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const userinfo = () => {

return Mock.mock({

"ret": 0,

"message": "获取用户信息成功",

"data": [{

"user\_id": 38,

"username": "190",

"user\_account": "190",

"password": "2397346b45823e070f6fc72ac94c0a999d234c472479f0e26b30cdf5942db854",

"phone": "190",

"create\_time": "2021-06-14T05:04:06.000Z",

"update\_time": "2021-06-14T05:04:06.000Z",

"receive\_address": "190"

}]

})

}

const deleteAddress = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const all = () => {

return Mock.mock(

{

"ret": 0,

"data|1-10": [{

"product\_id|+1": 1,

"product\_name": "智能手机",

"product\_desc": "爆款手机",

"product\_type": "电子",

"price": "200.00",

"stock": 47,

"image": "https://tse2-mm.cn.bing.net/th/id/OIP.gAJgq18dd5Dgn2SjLRwX3QHaHa?w=172&h=180&c=7&o=5&dpr=1.25&pid=1.7",

"create\_time": "2021-06-13T08:51:26.000Z",

"update\_time": "2021-06-13T08:51:26.000Z"

}, {

"product\_id": 2,

"product\_name": "智能手机",

"product\_desc": "爆款手机",

"product\_type": "电子",

"price": "300.00",

"stock": 10,

"image": "https://tse3-mm.cn.bing.net/th/id/OIP.6CnBqwxGBzSfw2DzKlxXBQHaO0?w=115&h=180&c=7&o=5&dpr=1.25&pid=1.7",

"create\_time": "2021-06-13T08:51:30.000Z",

"update\_time": "2021-06-13T08:51:30.000Z"

}, {

"product\_id": 3,

"product\_name": "老年机",

"product\_desc": "性能手机",

"product\_type": "电子",

"price": "50.00",

"stock": 195,

"image": "https://tse3-mm.cn.bing.net/th/id/OIP.lw2EnlG03aj4y3PNMsntrgHaHW?w=179&h=180&c=7&o=5&dpr=1.25&pid=1.7",

"create\_time": "2021-06-13T07:09:36.000Z",

"update\_time": "2021-06-13T07:09:36.000Z"

}]

}

)

}

const getMyCart = () => {

return Mock.mock(

{

"ret": 0,

"message": "成功",

"data": [{

"cart\_id": 307,

"product\_name": "智能手机",

"price": "200.00",

"amount": 1,

"product\_desc": "爆款手机",

"image": "https://tse2-mm.cn.bing.net/th/id/OIP.gAJgq18dd5Dgn2SjLRwX3QHaHa?w=172&h=180&c=7&o=5&dpr=1.25&pid=1.7"

}, {

"cart\_id": 308,

"product\_name": "智能手机",

"price": "200.00",

"amount": 1,

"product\_desc": "爆款手机",

"image": "https://tse2-mm.cn.bing.net/th/id/OIP.gAJgq18dd5Dgn2SjLRwX3QHaHa?w=172&h=180&c=7&o=5&dpr=1.25&pid=1.7"

}]

})

}

const addToCart = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const deleteMyCart = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const getMyAddress = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

const logout = () => {

return Mock.mock({

...{

ret: 0,

msg: "成功",

code: 0

}

})

}

Mock.mock('/user/login', login)

Mock.mock('/user/register', register)

Mock.mock('/user/addAddress', addAddress)

Mock.mock('/user/userinfo', userinfo)

Mock.mock('/user/deleteAddress', deleteAddress)

Mock.mock('/product/all', all)

Mock.mock('/cart/getMyCart', getMyCart)

Mock.mock('/product/addToCart', addToCart)

Mock.mock('/cart/deleteMyCart', deleteMyCart)

Mock.mock('/user/getMyAddress', getMyAddress)

Mock.mock('/user/logout', logout)