与顾客app登陆有关的设置：

1.在router/index.js路由配置上需要有拦截：（只有登陆成功才可以显示数据）

//管理页面路由配置

{

path: '/manager',

    name: 'manager',

    component: Manager,

    beforeEnter: (*to*, *from*, *next*) => {  *//*属于路由自己的拦截器

      let token = getToken();

      if(token){

*//* 查询*info(状态机分发)*

        store.dispatch('user/info',token)

        .then(()=>{

*//* 当获取万用户信息之后才允许跳转

          next();

        })

      } else {

        Toast("token失效")

*//* 跳转到登录

        next({path:'/login'})

      }

},

子页面路由配置

Children:[{

...

}]

},

//同级登陆注册等页面路由配置

{

path: '/login,

name: login,

...

},

2.在store/modules/usr.js中开始访问接口查询数据

import {get,post\_json,post} from ‘@/http/axios'

//使用vue框架自带的组件，用以设置，获取，移除token的值。

import {setToken,getToken,removeToken} from '@/utils/auth'

export default {

  namespaced:true,

  state: {

    cusInfo:{},

    token:getToken(),

    info:{} *//*用户信息

  },

  mutations: {

    refreshCusInfoById(*state*,*cusInfo*){

*state*.cusInfo = *cusInfo*

    },

    refreshInfo(*state*,*info*){

*state*.info = *info*;

    },

    refreshToken(*state*,*token*){

*state*.token = *token*;

    }

  },

  actions: {

*//* 根据顾客*id*获取顾客信息

    async FindCustomerById({*commit*},*id*){

      let response = await get('/customer/findCustomerById?id=' + *id*)

      commit('refreshCusInfoById',response.data)

    },

*//* 登录

    async login(*context*,*payload*){

*// 1.*登录

      let response = await post\_json("/user/login",*payload*);

*// console.log("payload",payload)*

      let token = response.data.token;

*// 2.*将*token*缓存起来？*localStorage*

      setToken(token);

*// 3.* 将*token*维护到状态机中

*context*.commit("refreshToken",token);

    },

*//* 通过*token*获取*info*

    async info(*context*,*token*){

*// console.log('info');*

      let response = await get("/user/info",{token});

*//* 将用户信息设置到*info*中

      await *context*.commit("refreshInfo",response.data)

*// console.log('info-end');*

    },

*//* 退出

    async logout(*context*){

*// 1.* 请求后台退出

      await post('/user/logout');

*// 2.* 清理本地缓存

      removeToken();

*// 3.* 清理*store*

*context*.commit('refreshToken','');

*context*.commit('refreshInfo',{});

    }

  }

}

3.登陆页面的vue代码：

<template>

  <div class="login">

    <div class="header">

      <div class="logo">家</div>

      <div class="title">

        家政服务云平台

        <span style="font-size:.8em">顾客端</span>

      </div>

    </div>

    <div class="form">

      <van-cell-group>

        <van-field

          v-model="form.username"

          required

          clearable

          label="用户名"

          icon="question-o"

          placeholder="请输入用户名"

        />

        <van-field

          v-model="form.password"

          type="password"

          label="密码"

          placeholder="请输入密码"

          required

        />

      </van-cell-group>

      <div style="text-align:center;padding:1em" >

       <van-button

        plain round

        @click="loginHandler"

        style="width:100%">登录</van-button>

      </div>

    </div>

  </div>

</template>

<script>

import {mapActions} from 'vuex'

export default {

  data(){

    return {

      form:{

        type:'customer'

      }

    }

  },

  methods:{

    ...mapActions("user",["login"]),

*//* 登录

    loginHandler(){

      this.login(this.form)

      .then(()=>{

*//*跳转到首页

        this.$router.push({path:'/manager/home'})

      })

    }

  }

}

</script>

<style scoped>

.register{

  margin:1em;

  font-size:12px;

  color:green;

}

.register button{

  border:none

}

.header {

  height: 140px;

  padding: 50px;

  color: #ffffff;

  background-color: #1659a0;

  text-align: center

}

.header .logo {

  width: 50px;

  height: 50px;

  line-height: 50px;

  border-radius: 50%;

  border: 1px solid #efefef;

  margin: 0 auto;

  color:#efefef;

  font-weight: bold;

  margin-bottom: 1em;

  font-size: 1.5em

}

.header .title {

  font-size: 24px;

}

</style>