外卖后台管理系统

1. 项目介绍

外卖管理系统这个项目是面向店铺商家的，主要功能是为商家提供订单统计和销售额统计的数据可视化，订单的管理，商品管理，店铺管理，账号管理等在线服务，让所有业务数据化，可操作。

1. 功能模块

**本系统功能模块包括**：

**首页**：显示总订单，总销售额，今日订单，今日销售额以及订单和销售额关于时间的可视化折线图

**订单管理**：展示店铺的所有订单，且商家可对每条数据进行查看和编辑，表格底部还提供分页功能，可显示订单总条数和跳转页码，除了这些，还可以通过订单号，收货人，手机号，订单状态和下单时间对订单进行模糊查询

**商品管理**：分为商品列表，商品添加和商品分类。同样，商品列表也可通过商品名称和分类进行模糊查询，可对每一个商品进行编辑和删除操作，表格下方提供了完整的分页功能，可显示数据总条数，可自定义每页显示条数和跳转到指定页。商品添加页面可实现商品名称，价格，分类，图片和描述自定义。添加商品成功，自动跳转到商品列表页。商品分类可实现商品分类名称自定义和是否启用，也可以添加新的商品分类。商品分类名称与商品添加页面的商品分类选项一一对应

**店铺管理：**默认状态为禁用，点击编辑可自定义所有店铺信息，支持店铺头像上传和店铺照片墙上传。

**账户管理：**分为添加账号，账号列表，修改密码和个人中心。账号列表可实现对账号的编辑，删除，批量删除和取消 全选。个人中心支持用户头像的上传和修改。

**销售统计：**分为订单统计和商品统计的可视化图表，都支持基于时间选择器的模糊查询功能

1. 技术栈。

HTML+CSS+Less+JavaScript+Vue.js+Element UI+Axios+Node.js+echarts+Git

1. 项目难点

**面包屑：**通过this.$route.matched计算出面包屑数组，使用element的bread-crumb插件将面包屑数组循环遍历出来。

**拦截器：**统一在请求拦截中为请求头携带token，在响应拦截中监听code，统一提示成功和失败信息

**管理员权限：**实现不同身份的管理员可访问的内容不同，不可访问的内容直接不可见。通过当前登录的账号的role和动态路由计算出当前账号可访问的路由，最后使用vue-router的addRoutes方法将路由动态挂载。除此之外还需通过可访问的路由计算出可访问的菜单，最后将菜单动态渲染到导航栏。

1. 问题及解决

·面包屑功能只有刷新和进入才正常切换

解决：watch: {

"$route.path"() {

this.calcBread();//忘记写this，导致函数无法正常调用

}

·登录组件中，点击登录没有触发http请求

问题：没有正确暴露登录ajax函数

/\* 登录请求 \*/

export function checkLogin(params) {

    return req.post("/users/checkLogin", params)

}

·用户列表组件，批量删除操作无法获取到被选中的行

解决：这里运用了表格的change-selection事件，主要分清事件和方法

@selection-change="handleSelectionChange"

 // 当选择项发生变化时会触发该事件

    handleSelectionChange(rows) {

      this.ids = rows.map(v => v.id);

      console.log(this.ids);

    },

·头像或单张图片上传/更新问题

头像/图片上传成功回调返回的imgUrl为xxxx.jpg，图片直接回填到xxxx.imgUrl

修改头像/图片请求参数的imgUrl为xxxx.jpg，上传时，直接使用xxxx.imgUrl

显示头像/图片的方式为，显示上传需要设置src为imgBaseUrl+xxxx.imgUrl

 <img v-if="xxxx.imgUrl" :src="imgBaseUrl+xxxx.imgUrl" class="avatar" />

说明：

如果响应的图片数据为xxx.jpg，那么上传回填就用xxx.jpg

如果响应的图片数据为<http://......jpg>，那么上传回填就用imgBaseUrl+xxx.jpg

·店铺管理的照片墙问题

1.获取店铺信息照片墙数据格式:

pics:[xxx.jpg,xxx.jpg,xxx.jpg]

2.照片墙显示方式:

:file-list//属性值为[{name: 'food.jpg', url: 'https://xxx.cdn.com/xxx.jpg'}]

因此在1中获取到pics，需要进行数据处理

let { data } = await getShopInfo();

    this.shopForm = data;

    this.shopForm.pics = this.shopForm.pics.map(v => ({

      name: v,

      url: this.imgBaseUrl + v

    }));

3.上传照片到照片墙===>对接修改店铺信息的照片墙请求参数

handleImgSuccess(res) {

      let { code, msg, imgUrl } = res;

      if (code === 0) {

        this.$message({ type: "success", message: msg });

        this.shopForm.pics.push({

          name: imgUrl,

          url: this.imgBaseUrl + imgUrl

        });

      }

    }

4.从照片墙删除照片===>对接修改店铺信息的照片墙请求参数

handleRemove(file) {

      this.shopForm.pics.forEach((v, i) => {

        if (v.uid === file.uid) {

          this.shopForm.pics.splice(i, 1);

        }

      });

    },

5.发送修改店铺信息请求时，还需要将pics进行处理

因为当前pics的格式为{name:”xxx.jpg”,url:<http://.....>,uid:"153365”}

**let** arr = this.shopForm.pics.map(v **=>** v.name);

        pics: JSON.stringify(arr)

注意：请求参数如果时数组，还需要用JSON转成字符串，才能成功发送请求。

· Ajax请求参数有时间选择器数组时，需要判断请求参数为null的情况，如果为null，设置参数为”[]”