**前台项目登录逻辑**：

用户的状态有2种：

1.临时Id；用来在用户未登陆时，用于保存购物车的保存信息，临时id随着请求发送保存在请求头中；

2.账号登陆：登录时向对应地址发送ajax请求需要携带账号和密码，在发送前需要对数据进行验证；登录成功，将请求到的token令牌通过请求拦截器放到请求头中，同时通过本地存储保存，这样每次刷新浏览器就不会需要重新请求登录直接通过本地存储获取，如果此时存在临时id，则将临时id中购物车的数据剪切到用户的购物车数据中；

**配置全局前置路由守卫**：

配置token校验

1.判断有没有token, 如果没有则暂时先放行（在需要在用户信息的页面要进行拦截跳转到登录页）

2.如果token已经存在，如果是去往登录页，则直接回到主页，如果不是则继续判断

3.token存在，并且不去登录页，则判断用户信息是否已经也存在；如果用户信息存在，则直接放行（此时已经登录并且用户名也存在）；如果用户信息不存在，则去请求用户信息，请求完成后再放行,如果请求失败，则统一认定token过期去再次登录，登录后展示用户名和退出登录，首页点击我的购物车跳转

4.去往登录的路径缓存，当通过路由守卫拦截跳转到登录页面时将当前要去往的路径地址保存到query中（在路由守卫中通过to.path获取），当登录成功时判断query是否存在缓存，存在则跳转到对应路径，否则直接回到主页

**前台项目购物车逻辑**：

1. 获取购物车列表 向/api/cart/cartList接口发送GET请求，此时返回一个数据为空数组，

2通过第三方库”uuid”添加标识，发送请求时通过拦截器给每个axios请求添加上请求头的代表身份的id，id要唯一，不能随意修改，再vuex数据初始化时生成了标识作为id并通过localStorage存储在本地，通过判断本地是否存在id来决定是否生成新的id作为用户标识，服务端通过对id的对比返回改用户添加到购物车的数据

3.对购物车已有物品进行数量改动，向/api/cart/addToCart/{ skuId }/{ skuNum }接口发送POST请求，skuId为商品ID，skuNum商品数量正数代表增加负数代表减少（直接修改数量时则发送数量为：修改前后数量的差值，若修改后数量小与1则视为商品数量改为1），为若成功返回一个数据为null，code为200的数据代表请求发送完成对数据进行了改动，此时再发送请求重新获取数据对购物车详情页面重新渲染

4.切换商品选中状态， 向/api/cart/checkCart/{skuID}/{isChecked}接口发送GET请求，skuId为商品ID，isChecked为改变后的选中状态0代表取消选中1代表选中，此时返回一个数据为空数组，为若成功返回一个数据为null，code为200的数据，此时再发送请求重新获取数据对购物车详情页面重新渲染

5.全选切换商品选中状态， 向/api/cart/batchCheckCart/{isChecked}接口发送Post请求，isChecked为改变后的选中状态0代表取消选中1代表选中，需要参数skuIdList为数组，值修改的商品id列表(通过foreach遍历数据将每个数据的id通过push方法添加到skuList数组中)为若成功返回一个数据code为200的数据，此时再发送请求重新获取数据对购物车详情页面重新渲染

6. 删除购物车商品， 向/api/cart/deleteCart/{skuId}接口发送DELETE请求，skuId为需要删除的商品ID，若成功此时返回一个数据为空数组，为若成功返回一个数据为null，code为200的数据，此时再发送请求重新获取数据对购物车详情页面重新渲染

7.批量删除购物车商品， 向/api/cart/batchDeleteCart接口发送get请求，需要参数skuIdList为数组，值修改的商品id列表(通过reduce遍历数据对比数据上携带的isChecked的值若为1则通过push方法添加到skuList数组中)请求体参数存放在配置对象的data上，，为若成功返回一个数据code为200的数据，此时再发送请求重新获取数据对购物车详情页面重新渲染，若当全部删除后，全选按钮是选中状态则需要通过判断this.shopCartList.length是否为0来确定全选按钮的选中状态

**前台项目支付逻辑**：

1.交易页面：进入到交易页面需要获得两类数据：1：收件人信息，2：商品清单。通过ajax请求获取到信息后并渲染到页面，对页面需要提交的数据进行双向绑定;通过拿到的tradeNo和tradeInfo去请求订单号，点击提交订单时请求;请求成功：获取到支付的订单号，并通过路由传参携带到支付页面；请求失败：给出提示并什么都不用做，如果没有库存也是请求失败

2.支付页面：跳转到支付页面后通过支付的订单号会获取到一些数据，比如支付二维码和订单号;点击立即支付时会弹出一个ElementUI组件的弹窗，并要用qrcode的包去把支付二维码解析成图片格式然后放在弹窗中，通过**轮询**不间断的向服务端发送请求，就会得知你的支付状态；会有两个按钮：我已支付和遇到问题，当点击遇到问题时弹出请联系客服之类的提示，当点击我已支付发送核对支付的ajax请求，请求成功和失败做出相应的提示，成功后跳转；付款成功页面：配置独享路由守卫，支付成功页面必须由支付页面跳转，支付页面必须由交易页面跳转。

**平台属性、三级联动**

后台项目平台属性的逻辑：平台属性的类容需要分成两个组件

三级联动组件：

1.进入页面就要通过ajax获取到一级分类列表的信息，并渲染到一级分类的列表中；通过ElementUI的el-select组件携带的change事件和双向绑定去实现一级分类列表的监听，得知点了和点了哪一个

2.点击了一级分类列表后，通过拿到的ID值去请求二级分类列表的信息并渲染；通过以上的步骤去得到三级分类列表的信息并渲染，这样就可以实现三级联动了

3.数据的清空：但点击一级分类列表时需要去清空二级，三级分类列表的信息和数据，点击二级分类列表时清空三级分类列表的信息和数据

4.在三级列表选中值时，触发父组件绑定的自定义事件，实现向父组件发送属性列表的请求配置信息，点击一、二、三级分类列表时都要调用事件并传递一个对象，因为后面父组件发请求需要用到我们选中的三个id，所以在这我们要把对应选中的分类id传给父组件实现父组件的事实更新；

属性列表组件：

以为但页面切换时路由未跳转，所以需要通过v-show来实现选项卡效果

1.显示属性列表页：当点击到第三级分类时，需要通过自定义事件拿到的参数去发送获取属性列表的ajax请求，获取属性列表信息后渲染到页面中，先在data中初始化需要添加或更改需要的请求参数，点击增加属性按钮：需要跳转到添加页，但是只有三级分类列表被选中时，按钮才可点否则禁用，并要清空显示添加或修改页的数据，点击修改按钮，还是需要跳转到修改页，但此时需要把属性列表的属性深拷贝一份给请求参数，其他的就和增加属性的逻辑一样了

2.点击删除按钮：通过row.id属性拿到当前点击的id然后发送删除属性列表的ajax请求

3.显示添加或修改页：属性名：通过双向绑定数据；点击添加属性值按钮：需要用到占位思想，需要先在属性值数组中用push方法去添加一个对象，此时通过数据响应的原理去修改页面，重新绘制表格，然后在表格中写入空白的属性名的输入框，并用双向绑定去收集数据；

4.点击取消按钮跳转到显示属性列表页

5.点击保存：会通过搜集好的请求参数去发送添加/修改的ajax请求，添加/修改的ajax请求是一个接口，但是添加没有id，修改有id，点击操作里面的删除：弹出一个确认框，点击确认时遍历属性值数组删除当前点击的那一个

组件的书写先静(1.拆分组件:定义，注册，使用；2,组：挂载到APP3.渲:通过render函数渲染，在页面中展示出来)后动(1.初始化数据：将要渲染的数据初始化定义到data中；2.交互：书写交互逻辑)

属性的添加与修改5步1.收集数据；2.处理数据3.发送请求4.请求发送成功处理（操作失败与成功处理）；5请求发送失败处理

菜单权限，按钮权限，路由守卫，

菜单权限：需要全局路由导航守卫和动态添加路由一起做路由菜单权限控制

全局路由导航守卫：用户登录，获取token，通过token校验去获取用户信息的基本信息（name和avatar）以及权限信息（routes:此时为拥有权限的异步路由名字组成的数组，buttons)；

在路由跳转时作条件验证（决定是否能正常跳转），路由跳转的临时缓存，用户信息的请求

动态添加路由 ：首先需要将路由拆分为特定功能的部分(1,常量路由2，异步路由3.任意路由)，

当用户获取到权限信息后，通过过滤筛选将符合条件(判断路由名是否存在在权限信息中)的异步路由组成一个新的数组，

再将新的异步路由数组与常量路由，任意路由合并成一个新的路由对象用于去遍历显示菜单；

同时通过路由动态注册(添加)路由中(通过router.addRoutes方法(参数：过滤后的异步路由对象与任意路由对象组成的数组))，用与路由的正常跳转与使用；

按钮权限： 全局注册一个userButton方法 方法功能：用于查找数组中是否存在对应值，存在则返回true，否则反之；

按钮权限依赖于菜单权限，当拥有菜单权限才能去实现菜单下的按钮劝降；

为需要权限控制的按钮添加v-if=“值：调用事先书写的方法传递按钮对应的权限值通过对比权限判断来决定是否展示按钮”