



本地购物车-列表购物车



列表购物车-基础内容渲染



准备模版

绑定路由

渲染基础列表

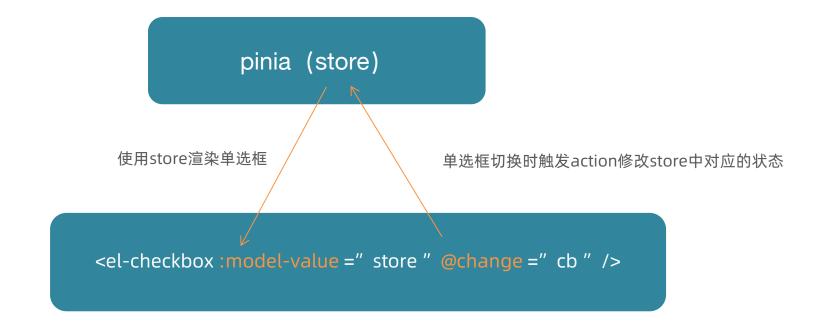


列表购物车-单选功能

核心思路:单选的核心思路就是始终把单选框的状态和Pinia中store对应的状态保持同步

注意事项: v-model双向绑定指令不方便进行命令式的操作(因为后续还需要调用接口),所以把v-model回退到一

般模式,也就是:model-value和@change的配合实现





列表购物车-全选功能

核心思路:

1. 操作单选决定全选:只有当cartList中的所有项都为true时,全选状态才为true

2. 操作全选决定单选: cartList中的所有项的selected都要跟着一起变

pinia (isAll ← cartList)

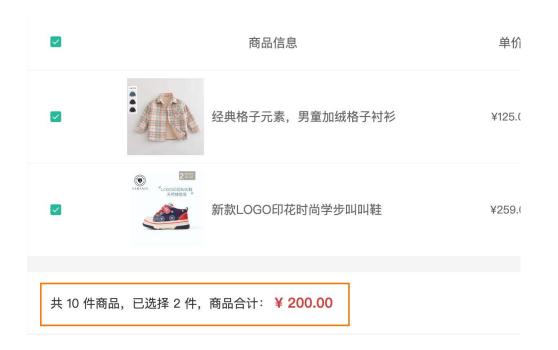
当store中cartList中的所有项都为true, isAll才为true

多选框切换时触发action把store中所有项的selected都修改为当前状态

<el-checkbox:model-value=" isAll " @change=" cb " />



列表购物车-统计数据功能



计算逻辑:

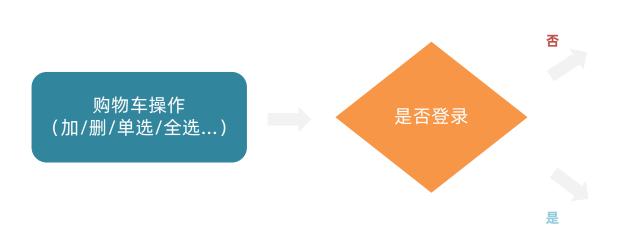
- 1. 已选择数量 = cartList中所有selected字段为true项的count之和
- 2. 商品合计 = cartList中所有selected字段为true项的count * price 之和







接口购物车-整体业务流程回顾



本地购物车操作

(所有操作不走接口直接操作Pinia中的本地购物车列表)

接口购物车操作

(所有操作直接走接口,操作完毕获取购物车列表更新本地 购物车列表)

结论

到目前为止,购物车在非登录状态下的各种操作都已经ok了,包括action的封装、触发、参数传递,剩下的事情就是在action中做登录状态的分支判断,补充登录状态下的接口操作逻辑即可



接口购物车-加入购物车

找到加入购物车的 action函数 判断是否 登录,如 果是

- 1. 调用加入购物车接口
- 2. 调用获取购物车列表接口
- 3. 用接口购物车列表覆盖本地购物车列表

```
const userStore = useUserStore()
const isLogin = computed(() => userStore.userInfo.token)
```



接口购物车-删除购物车

找到删除购物车的 action函数 判断是否 登录,如 果是

- 1. 调用删除购物车接口
- 2. 调用获取购物车列表接口
- 3. 用接口购物车列表覆盖本地购物车列表





退出登录-清空购物车列表



业务需求

在用户退出登录时,除了清除用户信息之外,也需要把购物车数据清空

在购物车store中补充 清除购物车的action userStore中找到退出 登录action,执行清除 业务





04 合并本地购物车到服务器



合并购物车业务实现

问:用户在非登录时进行的所有购物车操作,我们的服务器能知道吗?

答:不能!不能的话不是白操作了吗?还本地购物车的意义在哪?

解决办法: 在用户登录时, 把本地的购物车数据和服务端购物车数据进行合并操作

本地购物车

服务端购物车

实现步骤:

登录时调用合并购物车 接口

获取最新的购物车列表

覆盖本地购物车列表





结算模块-路由配置和基础数据渲染



路由配置和基础数据渲染

路由配置:

准备路由组件

配置路由关系

配置路由跳转

基础数据渲染:

准备接口

获取数据

渲染默认地址和商品列表 以及统计数据





结算模块-地址切换交互实现



地址切换交互需求分析

	切换收货地址 >	×
收货地址	收货人: 柴柴 联系方式: 13211111111 收货地址: 北京北京市东城区光明社区	
收货人: 小张 联系方式: 18771304318 收货地址: 湖南省长沙市芙蓉区 甜心咒	收货人: 老周 联系方式: 134111111111 收货地址: 北京北京市东城区昌平校区	切换地址
商品信息	收货人: 小张 联系方式: 18771304318	
商品信息	收货地址: 湖南省长沙市芙蓉区甜心咒	小计
抓绒保暖,毛毛虫儿童运动鞋26- 颜色:粉黑色(LXJ017) 尺码:27	取消 确定	¥327.00

1. 打开弹框交互: 点击切换地址按钮, 打开弹框, 回显用户可选地址列表

2. 切换地址交互:点击切换地址,点击确定按钮,激活地址替换默认收货地址



打开弹框交互实现

准备弹框组件

组件v-model绑定响应 式数据

点击按钮控制响应式数据

渲染可选地址列表



地址激活交互实现

收货人: 柴柴

联系方式: 13211111111

收货地址: 北京北京市东城区光明社区

收货人: 老周

联系方式: 13411111111

收货地址: 北京北京市东城区昌平校区

收货人: 小张

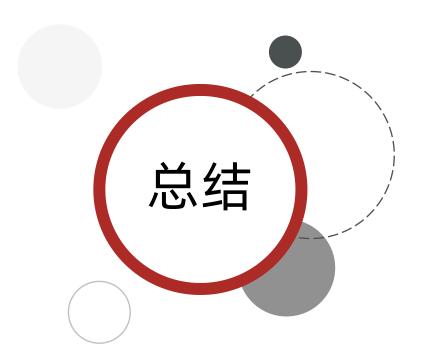
联系方式: 18771304318

收货地址:湖南省长沙市芙蓉区甜心咒

原理:地址切换是我们经常遇到的`tab切换类`需求,这类需求的实现逻辑都是相似的

- 1. 点击时记录一个当前激活地址对象activeAddress, 点击哪个地址就把哪个地址对象记录下来
- 2. 通过动态类名:class 控制激活样式类型 active是否存在,判断条件为:激活地址对象的id === 当前项id





切换地址属于哪类通用型交互功能?

tab切换类交互

记录激活项(整个对象/id/index) + 动态类名控制





订单模块-生成订单功能实现



业务需求说明

商品件数: 3件

商品总价: ¥327.00

运 费: ¥4.00

应付总额: 331.00

提交订单

确定结算信息没有问题之后,点击提交订单按钮,需要做以下俩个事情:

- 1. 调用接口生成订单id,并且携带id跳转到支付页
- 2. 调用更新购物车列表接口,更新购物车状态

准备支付页路由

封装生成订单接口

点击按钮调用接口,得到订单id, 携带id完成路由跳转





07 支付模块-渲染基础数据



渲染基础数据



应付总额: ¥385.00

支付页有俩个关键数据,一个是要支付的钱数,一个是倒计时数据(超时不支付商品释放)

封装获取订单详情的接口

获取关键数据并渲染

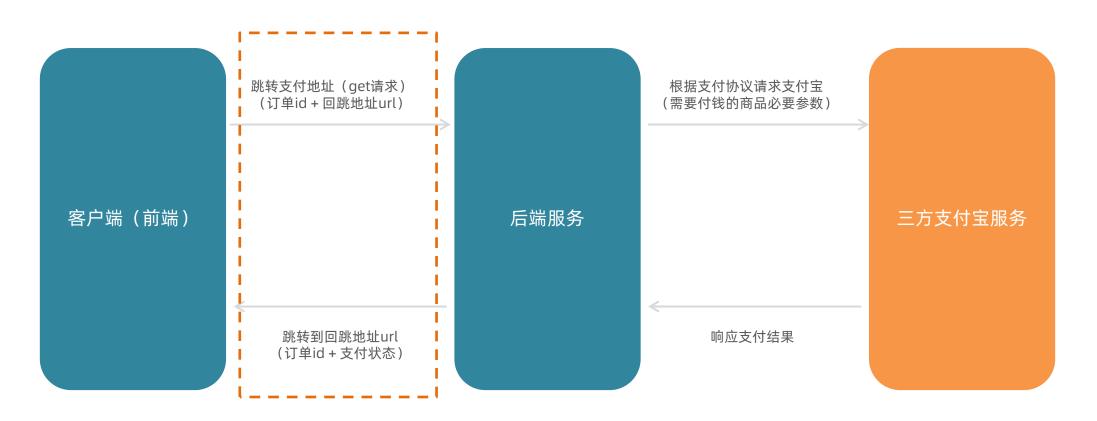




08 支付模块—实现支付功能



支付业务流程







支付模块-支付结果展示



业务需求理解



订单支付成功

我们将尽快为您发货, 收货期间请保持手机畅通

支付方式: 在线支付

支付金额: ¥256

查看订单

进入直页

○ 温馨提示: 小兔鲜儿不会以订单异常、系统升级为由要求您点击任何网址链接进行退款操作,保护资产、谨慎操作。

配置paycallback路由

根据支付结果适配支付状态

获取订单数据渲染支付信息





支付模块-封装倒计时函数



理解需求

```
▼ result: {id: "1630443796598624258", createTime: "2023-
arrivalEstimatedTime: null
  cityCode: "430100"
  closeTime: null
  consignTime: null
  countdown: 1712
  countyCode: "430102"
  createTime: "2023-02-28 13:44:28"
  deliveryTimeType: 1
  endTime: null
```



编写一个函数 useCountDown 可以把秒数格式化为倒计时的显示状态,函数使用样例如下:

```
const { formatTime, start } = useCountDown()

{ formatTime }}

start(60)
```

1. formatTime为显示的倒计时时间 2. start是倒计时启动函数,调用时可以设置初始值并且开始倒计时



实现思路分析

```
const { formatTime, start } = useCountDown()

{ formatTime }}

start(60)
```

1. formatTime为显示的倒计时时间 2. start是倒计时启动函数,调用时可以设置初始值并且开始倒计时

编写函数框架,确定参数和返回值

编写核心倒计时逻辑实现基础倒计时

实现格式化



传智教育旗下高端IT教育品牌