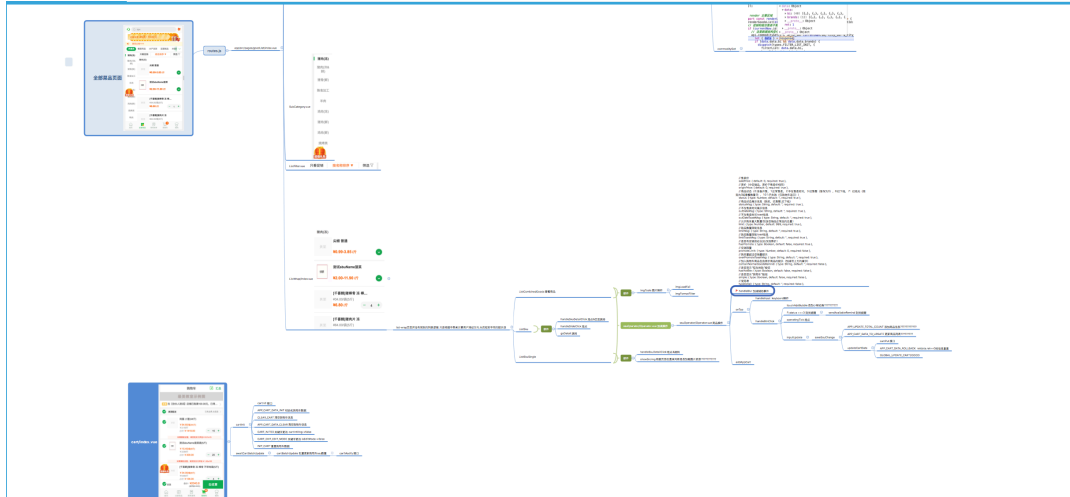


学习记录

1. 学习全部菜品页面 完善流程图



2. 学习Object.assign方法

3. 学习find 和 findIndex方法的使用

定义和用法

`findIndex()` 方法返回传入一个测试条件（函数）符合条件的数组第一个元素位置。

`findIndex()` 方法为数组中的每个元素都调用一次函数执行：

- 当数组中的元素在测试条件时返回 `true` 时, `findIndex()` 返回符合条件的元素的索引位置，之后的值不会再调用执行函数。
- 如果没有符合条件的元素返回 `-1`

注意: `findIndex()` 对于空数组，函数是不会执行的。

注意: `findIndex()` 并没有改变数组的原始值。

定义和用法

`find()` 方法返回通过测试（函数内判断）的数组的第一个元素的值。

`find()` 方法为数组中的每个元素都调用一次函数执行：

- 当数组中的元素在测试条件时返回 `true` 时, `find()` 返回符合条件的元素，之后的值不会再调用执行函数。
- 如果没有符合条件的元素返回 `undefined`

注意: `find()` 对于空数组，函数是不会执行的。

注意: `find()` 并没有改变数组的原始值。

已解决

1. `beforeRouteEnter` 什么时候回调用 是源页面跳转前执行？

目标页面访问前执行

```
const Foo = {
  template: `...`,
  beforeRouteEnter (to, from, next) {
    // 在渲染该组件的对应路由被 confirm 前调用
    // 不! 能! 获取组件实例 `this`
    // 因为当守卫执行前, 组件实例还没被创建
  },
  beforeRouteUpdate (to, from, next) {
    // 在当前路由改变, 但是该组件被复用时调用
    // 举例来说, 对于一个带有动态参数的路径 /foo/:id, 在 /foo/1 和 /foo/2 之间跳转的时候,
    // 由于会渲染同样的 Foo 组件, 因此组件实例会被复用。而这个钩子就会在这个情况下被调用。
    // 可以访问组件实例 `this`
  },
  beforeRouteLeave (to, from, next) {
    // 导航离开该组件的对应路由时调用
    // 可以访问组件实例 `this`
  }
}
```

`beforeRouteEnter` 守卫 不能 访问 `this` , 因为守卫在导航确认前被调用,因此即将登场的新组件还没被创建。

2. `appMethod`下的方法 例如`addShopCart`函数找不到 `appMethod`中的方法是在app端编写的方法本地代码无法查询 但是会根据app端提供的参数形式调用相应方法

```
    },
    addAppCart(count) {
      // APP 调用页面使用本组件时
      if (appMethod && appMethod.addShopCart && ['rmd'].filter(item => { return location.href.indexOf(item) > -1 })
        .length > 0) {
        // 这里从1.9.2版本的加入组合套餐开始 为了兼容老版本 SK_id = 0, unique_id, count
        appMethod.addShopCart(0, this.uniqueId, count);
      }
    },
    touchAddBubble($event) {
      let start = $event.type == 'touchstart' ? $event.changedTouches[0] : $event;
```

未解决

1. 加减号辅助事件有什么用？

```
// 加减号辅助点击事件
handleBlur($event) {
  setTimeout(function() {
    $event.target.blur()
  }, 100);
},
```

```
<template>
  <div class="ssu-operation-area">
    <!-- 简单模式只有加号 -->
    <div class="ssu-btn-area" v-if="simple">
      <a class="ssu-btn ssu-btn-plus"
        :class="{
          'ssu-btn-plus-disabled': btnDisabled || !goodsValid,
          'remind': P_status == 3
        }"
        v-tap="{ methods: onTap, num: 1 }"
        @touchmove.stop
        @touchend.stop="handleBlur"
        href="javascript:void(0);">
      </a>
    </div>

    <!-- 不可用, 显示信息 -->
    <div class="ssu-invalid" :class="{ ssu-invalid: {P_status}"
```