**1.通过路由配置解决非路由组件的显示与隐藏的问题**

这里的非路由组件是项目里面的放在底部的Footer组件，它们在某些页面会用到，通过配置每个路由的meta元数据属性，定义了一个控制footer显示与隐藏的属性，搭配v-show方法来控制footer的显示与隐藏

1. **对于轮播图组件中实例化 swiper 对象失败问题，使用监听属性和nextTick 解决**

造成这个现象的原因：  
 在含有轮播图的父组件挂载时发起了轮播图图片的异步请求，此时同步代码执行，但请求返回的数据还没有，此时不能渲染轮播图，实例化swiper对象无效。  
 解决方法：watch+nextTick  
 watch解决数据还没改变的问题：在轮播图组件（子）中监听轮播图的资源属性（父传子），当父传过来的属性变化，  
 this.$nextTick()解决数据传过来之后，v-for有可能还没执行完成的问题：在this.$nextTick函数中实例化swiper对象，此时轮播图对应的DOM结构已经生成。（nextTick在下次DOM更新之后，立即执行回调函数）

**3.鼠标移入商品分类列表过快导致浏览器卡顿的问题，使用节流解决（20ms）**

项目的首页有用到一个三级联动列表，对于一级列表，它是显示的，用户通过鼠标移动到对应一级列表，会展开其对应的二三级列表，而我们在一级列表上定义了回调函数，通过改变索引用来改变当前亮起的一级列表项。开始没有进行节流操作的时候，调试的时候发现鼠标移入商品分类列表过快会导致浏览器卡顿，

因此设置了一个节流函数，每20ms作为一个执行周期。

1. **对于点击一、二、三级列表中标签跳转到搜索页的步骤，利用事件委派和编程式路由实现**

这里事件委派的作用：减少工作量，通过事件冒泡捕获子元素的行为，在一级列表项定义了回调函数，在一二三级列表渲染的时候自定义了属性data-blabla。通过event.dataset对象获取触发跳转的元素以及它的自定义属性（分类名商品id什么的）。通过判断该元素是一级二级还是三级标签（因为从服务器请求回来的id是分级定义的），来拼接得到不同的路由参数。之后进行一个带参数跳转。

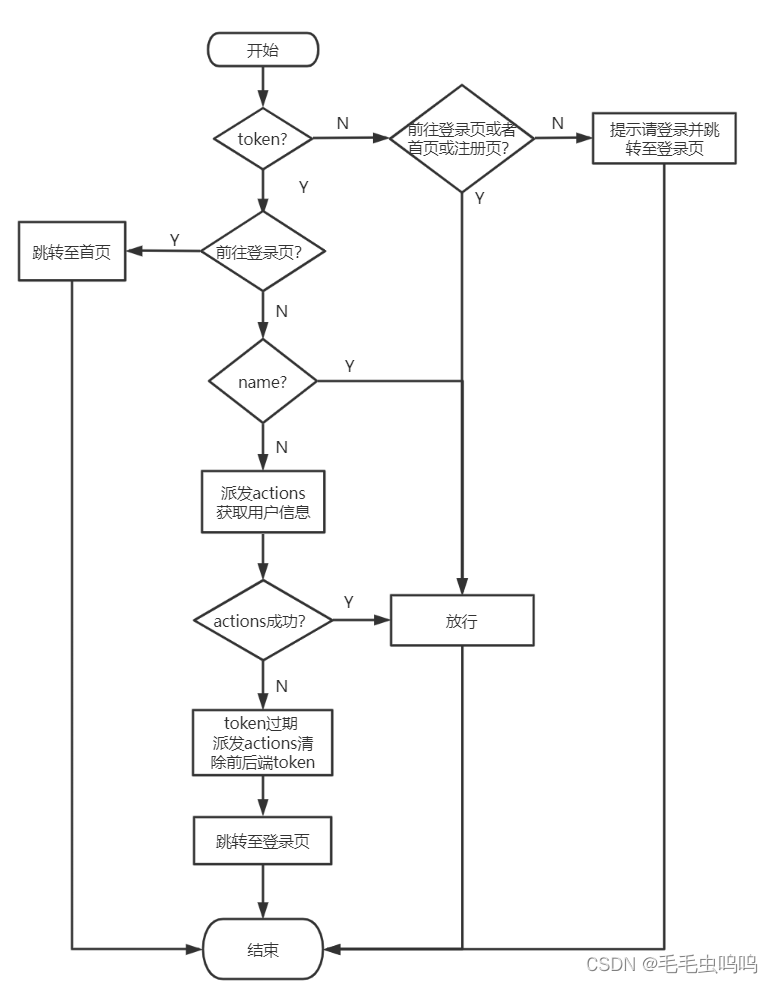
1. **针对公共组件重复发起异步请求的问题，利用根组件的生命周期函数进行性能优化**

这里的公共组件是三级列表组件，它在首页和搜索页都有用到，本来是在公共组件的mouted回调中进行请求，但是这样会导致切换不同页面时，反复发起异步回调，所以在根组件挂载时就发起请求，放入状态管理仓库。

1. **在购物车页面通过Promise的方法解决删除选中商品的多个异步请求**

通过Promise.all方法，将删除单个商品的异步请求函数返回的Promise对象放入数组中作为Promise.all的参数调用。

1. **使用导航守卫检测，用户未登录时进入其他页面会自动跳转登陆页面，登陆后自动跳转目标页面。**



1. **在订单页面通过 setInterval 定时器来轮询判断用户是否支付成功**

在支付页面点击支付后会弹出一个对话框，上面通过qrCode插件将该笔订单的支付页面转换成二维码形式，之后会启用一个定时器，不断向服务器查询当前订单的状态，通过返回的状态码决定是否清除该定时器以及跳转到支付成功页面