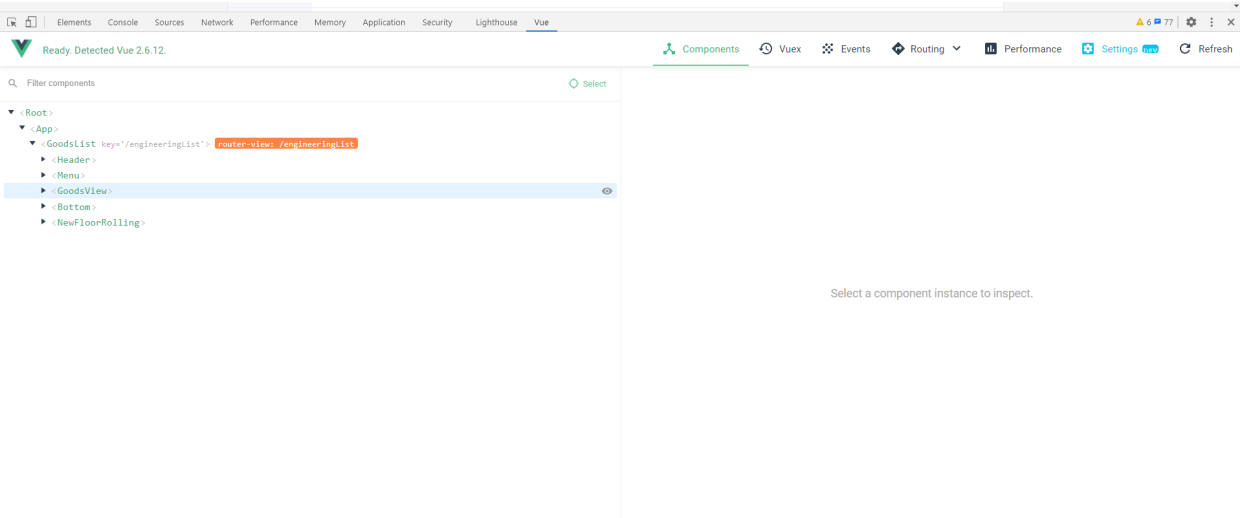


# Vue.js devtools面板

包含了：Components、Vuex、Events、Routing、Performance、Setting、Refresh



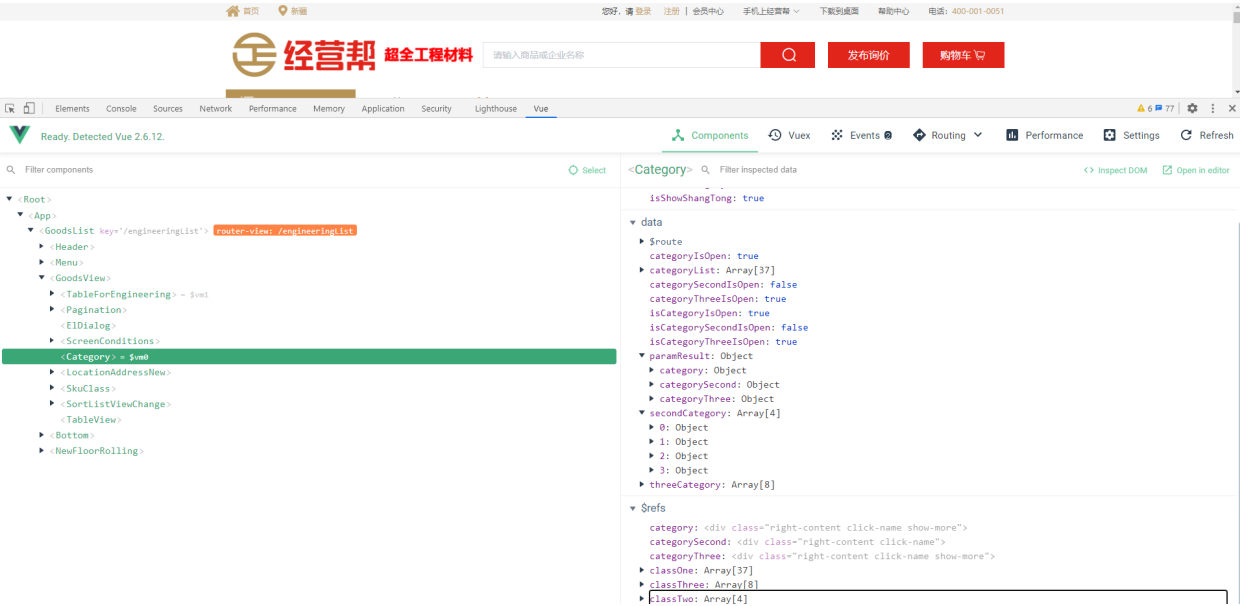
# Components面板

可以看到我们定义的一系列组件，选中相应的组件后，右侧面板可以看到组件内的data、props、computed等

左侧：页面包含的组件

右侧：组件对象的参数值，分别是：

- props：组件props传值
- data：组件内data对象包含的变量，\$route对象<path、query、params、fullPath、name、meta(配置项)>，同时data属性是可以编辑的，编辑后页面也会实时改变。
- ref：ref对象
- computed：只读不可编辑

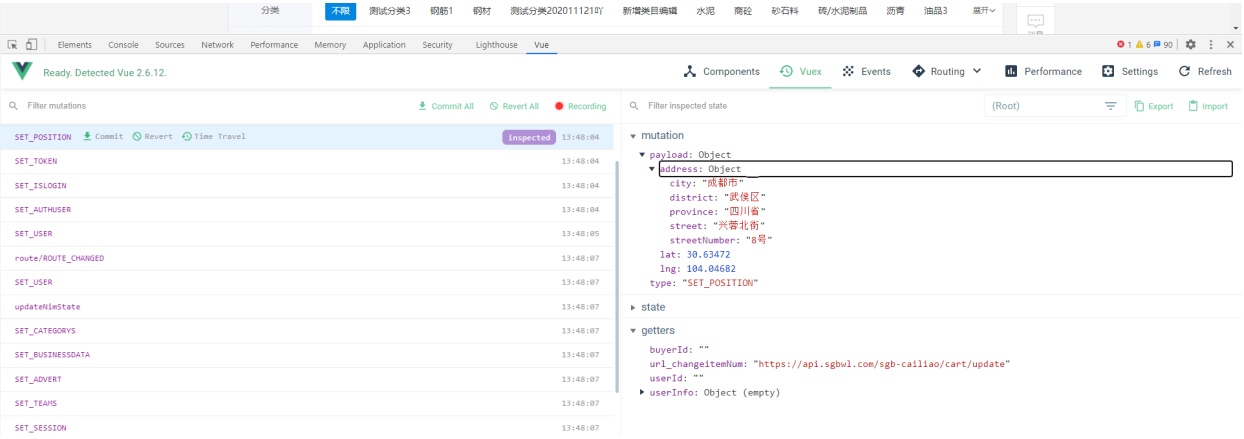


# Vuex

记录每一次mutation。点击对应的mutation记录在右边可以看到mutation提交的详细信息，跟此时state的状态。

鼠标滑过mutation记录会出现commit、revert、Time Travel。

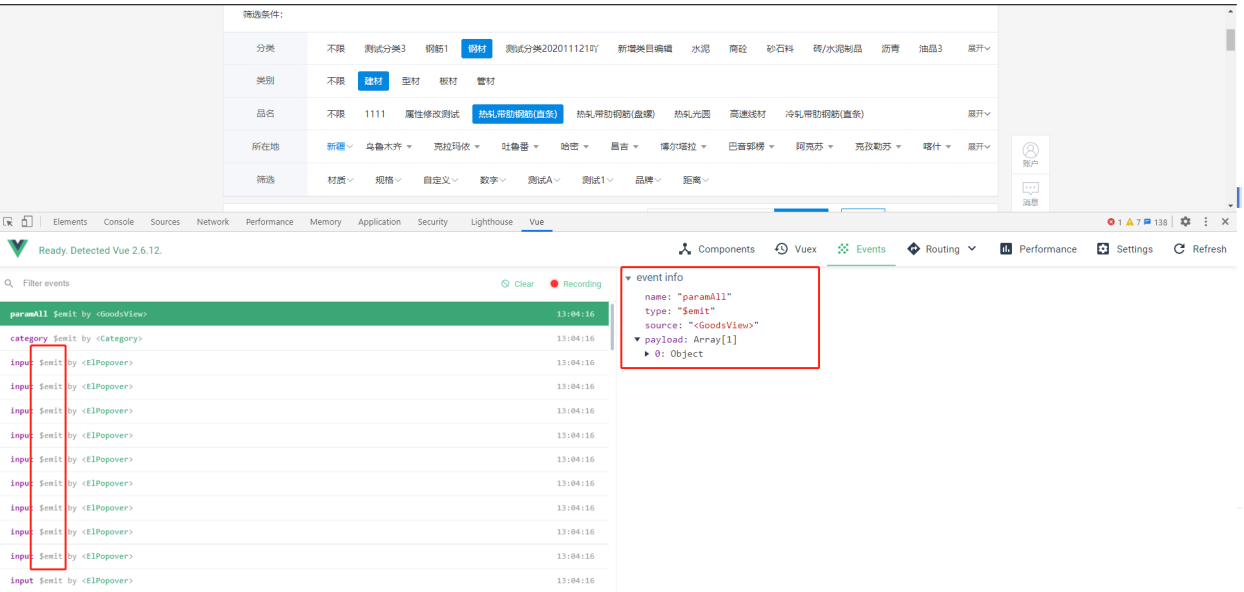
- commit: 保存指定的mutation，mutation记录最上面Base State会变为这次mutation的state，同时这次commit之前的mutation都会删掉。
- revert: 还原到指定mutation，当前的state变为指定mutation的state，revert之后的记录都会删掉。
- Time Travel: 时光旅行，state的数据切换到指定mutation，但所有记录都会保存。



## Events

记录每次\$emit触发的事件，右侧可以看到事件的详细信息

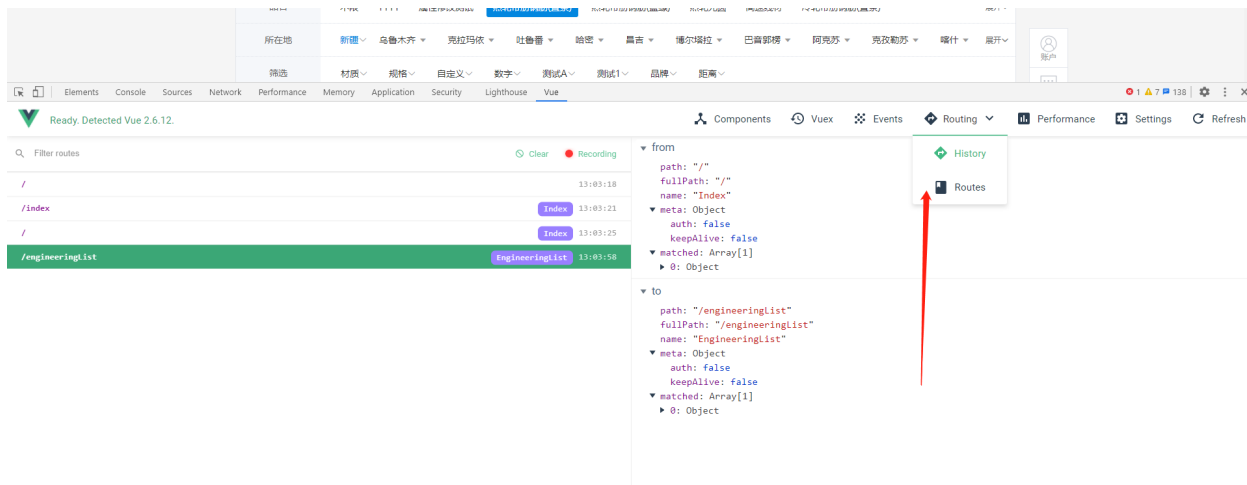
- name: 事件名
- type: 事件类型



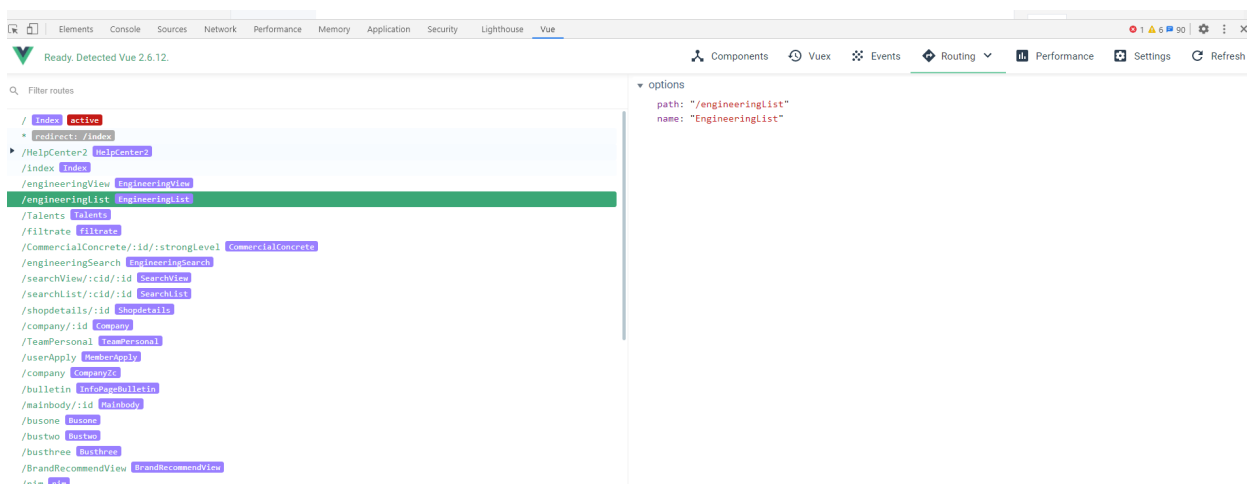
## Routing

面板中有个选项：*history*、*routes*。

- history：记录了每一次路由变化，但是注意这里的history !== window.history，每一次路由变化都会push到history里，即便你是this.\$replace跳转的路由。右侧会显示详细信息，面板当中的from、to跟vue-router路由钩子中from、to代表相同的意思



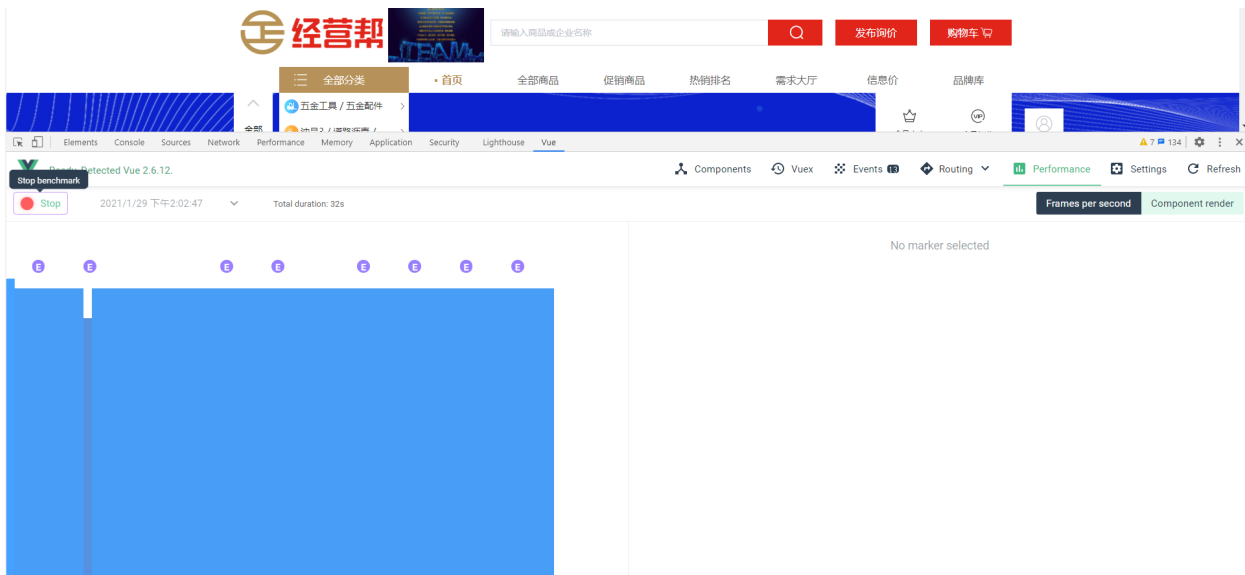
- routes：展示所有的路由，基本就是vue-router中routes的可视化展示



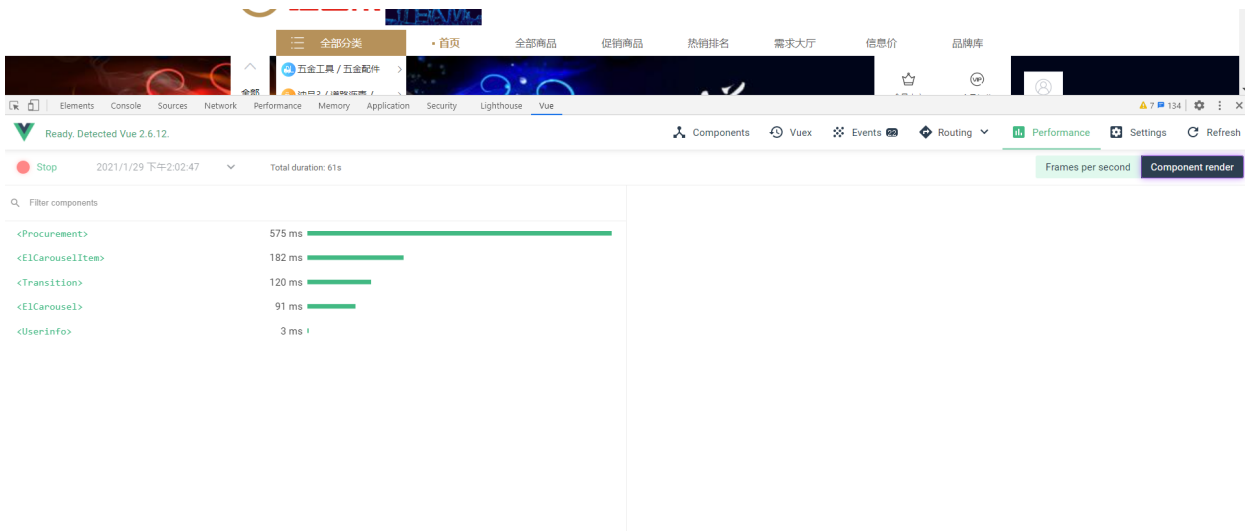
## Performance

可以帮助我们分析页面跟组件的渲染情况，它有两个功能 *Frames per second*、*Component render*。

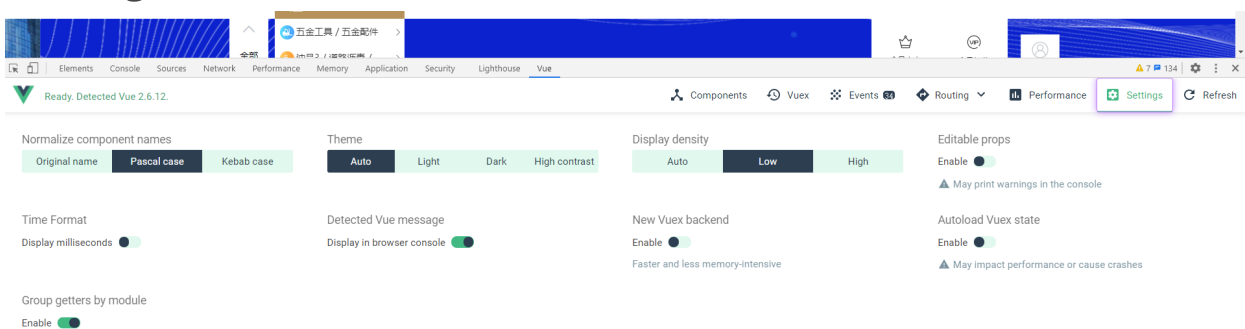
- Frames per second(fps)：浏览器每秒渲染多少帧，即每秒渲染多少次页面。蓝色柱体越高代表fps越高，代表页面越流畅。它同时可以记录引起页面渲染的原因，例如M、E、R。 M: mutation E: events R: routes



- Component render: 可以记录组件的渲染时间，注意这个时间是累加的，例如这个组件渲染了两次，那渲染时间是这两次的总时长。同时也可以看到组件内每个生命周期钩子的执行时长。



## Settings



Normalize component name: Components面板组件名字显示风格，origin name原始组件名，Pascal case大驼峰，Kebab case中横线。

Theme: 主题。 Time Format -> Display milliseconds: 时间格式显示到毫秒。 Autoload  
Vuex state: 是否自动加载vuex state

Editable props: 是否支持props属性可编辑，默认情况下components面板内的props属性不可编辑

New Vuex backend: 是否让Vuex面板单独在后台运行，更快且更少的内存占用。