# 项目笔记

Pages放路由 components放非路由

v-for(one in ones ) :key=“（唯一数值）”

Router-link标签 to跳转到路由

路由传参

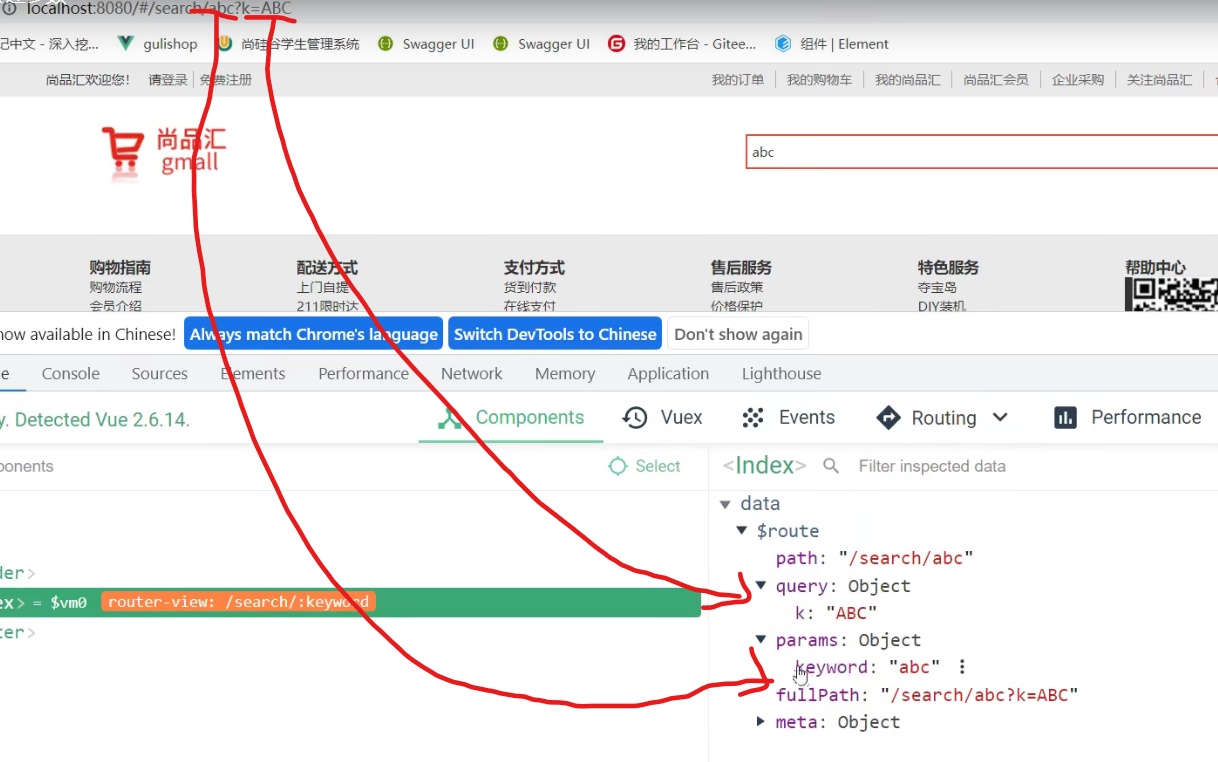
声明式导航：Router-link

编程式导航：利用$router.push|replace

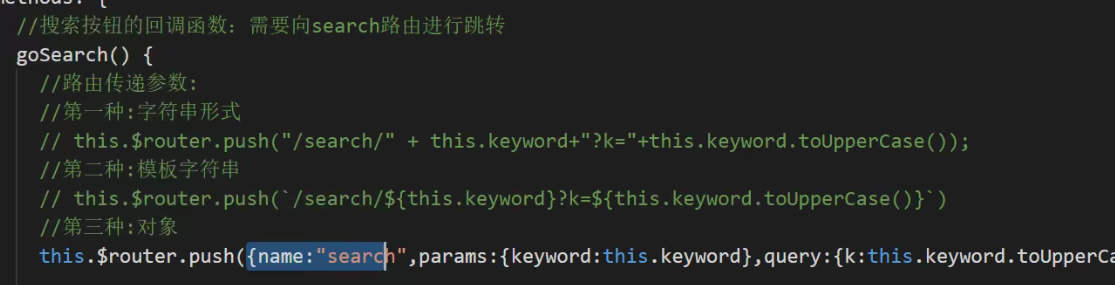
写法：

Params参数： 不属于路径一部分

Query参数：属于路径当中的部分



路由传参三种方式



路由配置文件路径占位:inputext

A页面传递参数到B页面

B路由文件上处理参数

3) 面试问题1:

描述: 编程式路由跳转到当前路由(参数不变), 会抛出NavigationDuplicated的警告错误

解决1: 在跳转时指定成功或失败的回调函数, 通过catch处理错误

解决2: 修正Vue原型上的push和replace方法 (优秀)

4) 面试问题2: 如何指定params参数可传可不传?

path: '/search/:keyword?'

5) 面试问题3: 指定params参数时可不可以用path和params配置的组合?

不可以用path和params配置的组合, 只能用name和params配置的组合

query配置可以与path或name进行组合使用

6) 面试问题4: 如果指定name与params配置, 但params中数据是一个"", 无法跳转

解决1: 不指定params

解决2: 指定params参数值为undefined

7) 面试问题5: 路由组件能不能传递props数据?

可以: 可以将query或且params参数映射/转换成props传递给路由组件对象

方法1.router组件js文件props:true

路由页面进行传递props:[‘变量名’]（只能传递params参数

方法2:props:{a:1,b:2}

方法3props: (route)=>({keyword1:route.params.keyword, keyword2: route.query.keyword })

全局组件注册方法：

在main.js里

导入组件

import TypeNav from '@/components/TypeNav'

全局组件 注册

Vue.component(TypeNav.name,TypeNav)

:data-xxxxxxxX(最后数据名字都会变成小写)

编程式页面跳转

router.push('home')

// 对象

router.push({ path: 'home' })

// 命名的路由

router.push({ name: 'user', params: { userId: '123' }})

// 带查询参数，变成 /register?plan=private

router.push({ path: 'register', query: { plan: 'private' }})

watch ：底层使用objectdefinepropty

对象写法:

BannerList:{

Handler(a,b){}

}

nextTick:在下次 DOM更新 循环结束（v-for）之后，执行回调，在修改数据之后 立即使用这个方法，获取更新后的DOM

ref用于获取dom，标签内：ref=”a”

this.$refs.a

组件间通信

Props:父子组件通信（三种写法

自定义事件：@on @emit 子给父传递

全局时间总线：$bus 全能

插槽 父子

Pubsub-js:完成

Vuex:万能

## Axios

## Vuex

Vuex官方提供的一个插件，状态管理库，集中式管理项目中组件共用的数据

应用于大项目，数据维护

State

Mutations

Actions

Getters

Modules

**基本使用**

建立仓库store

书写配置文件index.js

在文件中引入vue,vuex

使用vuex:Vue.use(Vuex)

State:用于存储数据

Mutations：修改state的唯一手段

Actions：处理action，可以自己书写业务逻辑，也可以处理异步

Getters：理解为计算属性，用于简化仓库数据，让组件获取仓库的数据更加方便

Mapstate辅助函数

使用vuex中的变量$store.state.nickname

页面显示麻烦，于是使用computed计算属性简写

computed:{

nickname(){

return this.$store.state.nickname

}

但是多个的时候书写起来麻烦所以用到mapstate函数

computed: mapState(['nickname','age','gender'])

**注意：用mapState等这种辅助函数的时候，前面的方法名和获取的属性名是一致的。**

修改别名使用：...mapState({bannerList:state=>state.home.bannerList})

主页Methods里派发方法：this.$store.dispatch(‘add’)

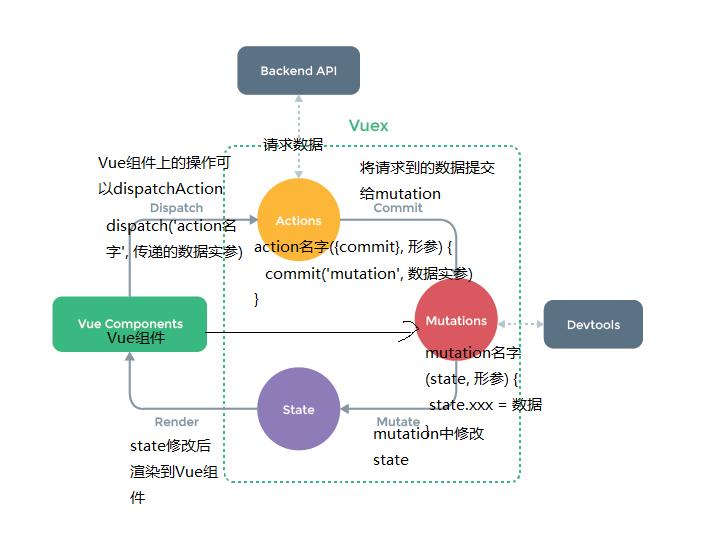
回仓库action里书写逻辑add({commit}){commit(“ADD”)}

Mutations里书写修改:ADD(state){state.count++}

Vuex当数据量过多时单独一个state难以维护数据，需要进行**模块开发**

在store里创建多个对应文件

在store创建相应文件夹，index.js进行配置，在store总配置文件进行引入，并声明模块



流程：

action中,

书写逻辑，将要在mutation中书写的函数名称作为参数写道commit中也可提交数据作为payload传递给mutation

Mutations中：

使用action中的函数名称和传过来的参数对state中的数据进行修改

...mapGetters(['名字'])

# 王元红/基础笔记

计算属性vs方法

计算属性有缓存，存储计算结果，当需要的data不变时，多次执行计算函数则会返回计算结果不执行函数

而方法则是每当触发重新渲染时，调用方法将**总会**再次执行函数。

Computed get set属性

fullName:{

get:function(){

}}

fullName:function(){

}

# 源码笔记

响应式：

Obj,defineproperty()

用于定义和修改对象属性，可读，可写等

Get函数（getter）

Set函数（setter）

虚拟Dom,diff算法

Vnode

Diff:h函数，patch函数