包含 拦截器的使用 + 参数配置,页面跳转,获取参数

1.Vue Router 是 Vue 官方的客户端路由解决方案。

客户端路由的作用是在单页应用 (SPA) 中将浏览器的 URL 和用户看到的内容绑定起来。当用户在应用中浏览不同页面时，URL 会随之更新，但页面不需要从服务器重新加载。

1. 安装

对于一个现有的使用 JavaScript 包管理器的项目，你可以从 npm registry 中安装 Vue Router：

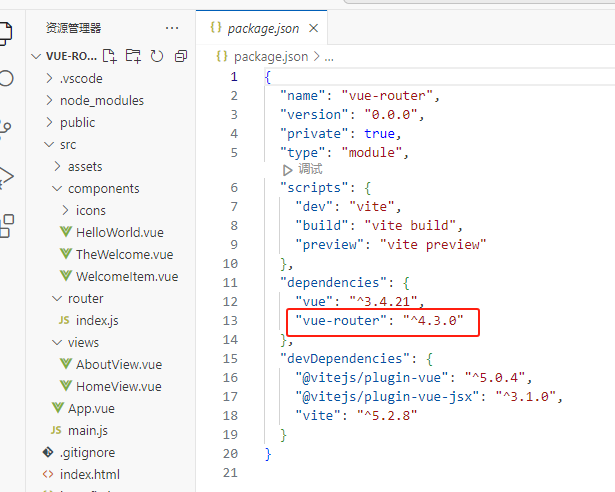
npm install vue-router@4

如果你打算启动一个新项目，你可能会发现使用 create-vue 这个脚手架工具更容易，它能创建一个基于 Vite 的项目，并包含加入 Vue Router 的选项：

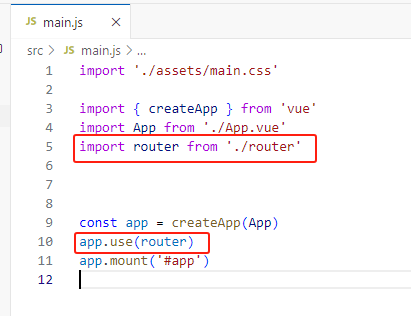
npm create vue@latest

1. 页面跳转

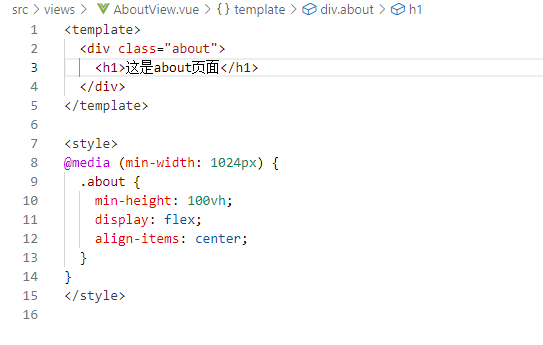
通过 npm create vue@latest命令得到的项目已经包含router：



在main .js中写入以下代码：



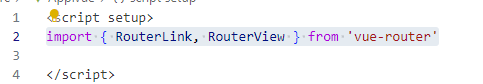
然后写跳转的页面：





在app.vue页面使用组件 RouterLink 创建链接，

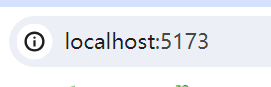
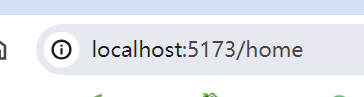
在js部分引用router



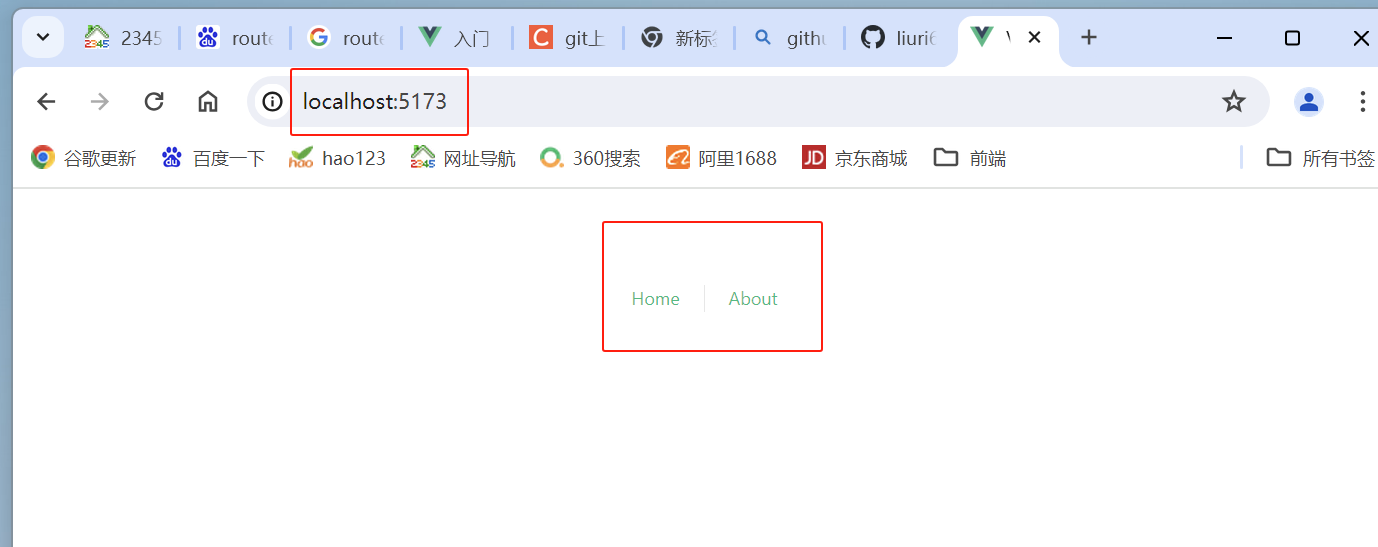
当点击时跳转到对应home和about的组件去，



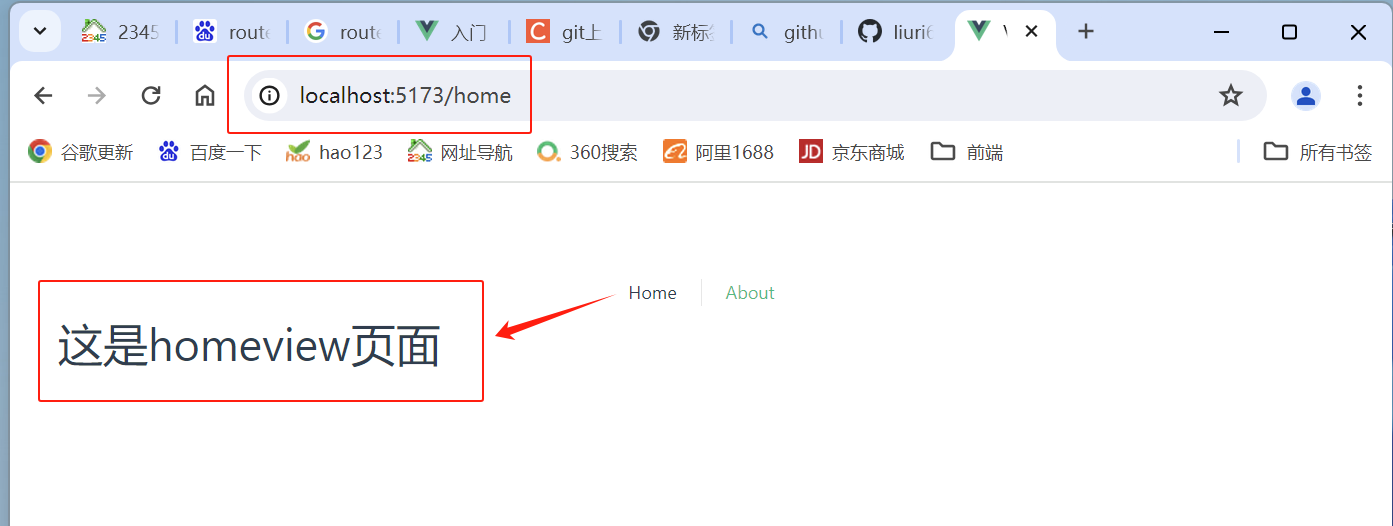
那怎么跳转到其他的页面呢，需要在项目的router文件夹下的index.js文件中写入以下代码：

home页面的路径就是主路径加上/home，比如我项目的路径是当我点击home链接之后，路径是这样的

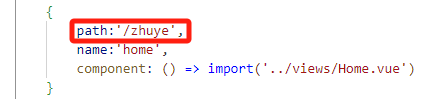
然后来看页面的显示：

根页面：

点击home之后路由跳转可以看到显示了homeview组件：



还可以在其他的地方写入： router.push("/zhuye") 这个是跳转到 [http://localhost:5173/zhuye](http://localhost:5173/zhuyede) 的url,跳转到对应的页面。/zhuye 对应index.js文件中的path：



或者写成这样也可以： router.push({ name: 'home' })也是可以的。

666666666666666666666666666666666666666666666666666666666666666666666666666666

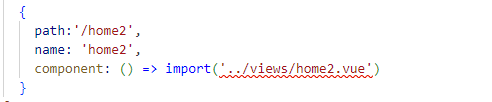
1. 路由跳转页面的几种方式

①RouterLink 见上面部分

②编程式导航 this.$router.push()

不带参数：

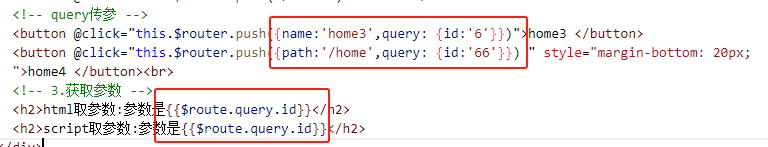




点击home2按钮就跳转到home2页面



query传参：

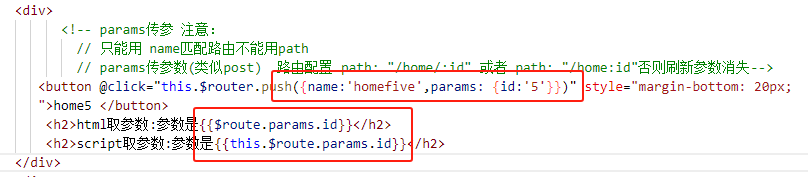




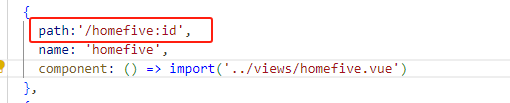
输出结果是取到了参数和跳转到对应的页面：



params传参：



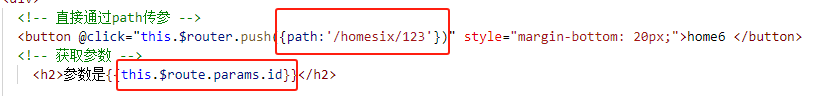
注意index文件的路径的写法：



显示结果是：



直接通过path传参：





页面结果是：

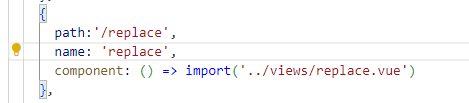


③ this.$router.replace()

用法同上：this.$router.push()

做一个简单的例子，其他的可以参考push的做法：





结果显示：

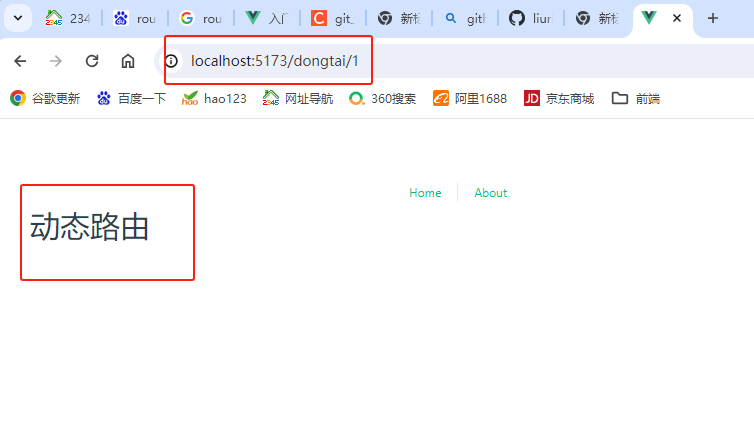


1. 动态路由传参：

在index.js 文件中写入以下代码：

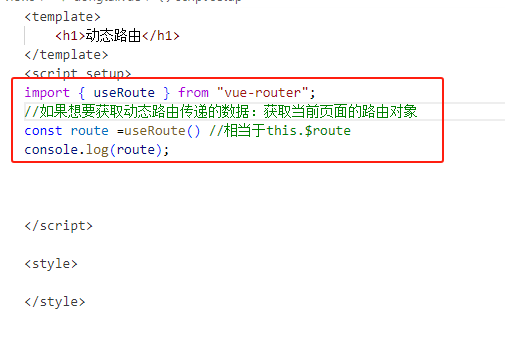


在浏览器接着输入/dongtai/1 可以跳转到动态路由的页面：



如果想要获取动态路由传递的数据：获取当前页面的路由对象

写入：

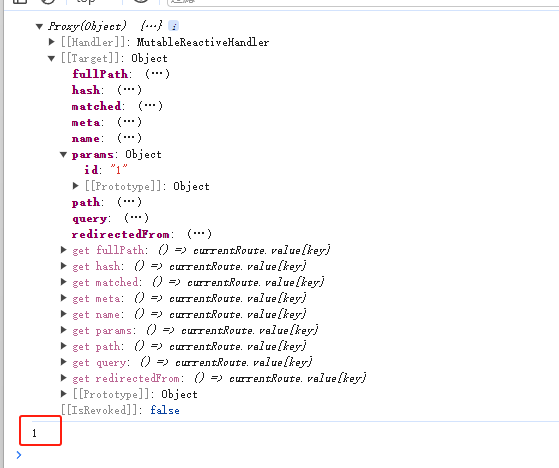


后台可以看到route的信息：



其中id的值是1；

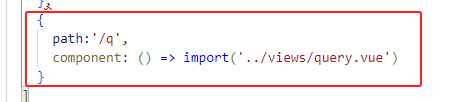
吧console.log(route) 改成 console.log(route.params.id);,来看输出的结果：



后台输出的是1 ，获取参数成功。

Query传参：

写入路由地址：



Query组件：



当在浏览器输入网址：

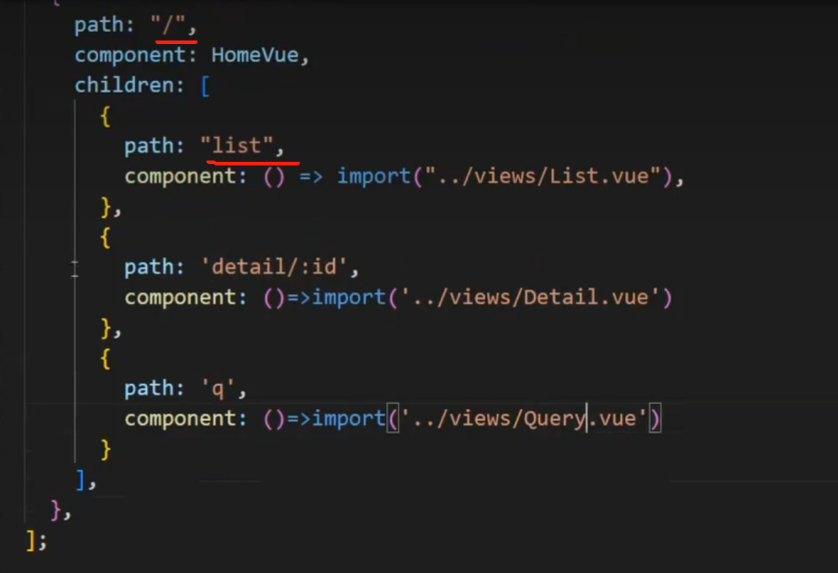
<http://localhost:5173/q?id=123> 说明传入的参数是123

页面输出：



获取成功

路由路径如果path定义了/ 下面就不用写了



方案四this.$router.go(n)

向前或者向后跳转n个页面，n可为正整数或负整数

query类似 get，跳转之后页面 url后面会拼接参数，类似?id=1。

非重要性的可以这样传，密码之类还是用params，刷新页面id还在。

params类似 post，跳转之后页面 url后面不会拼接参数。

区别:

1.this.$router.push

跳转到指定url路径，并在history栈中添加一个记录，点击后退会返回到上一个页面

this.$router.replace

跳转到指定url路径，但是history栈中不会有记录，点击返回会跳转到上上个页面 (就是直接替换了当前页面)

3.this.$router.go(n) this.$router.go(-1)===history.back() 返回上一页

向前或者向后跳转n个页面，n可为正整数或负整数

总结：

1.query可以用name和path匹配路由，通过this.$route.query.id获取参数，刷新浏览器参数不会丢失

2.params只能用name匹配路由，通过path匹配路由获取不到参数，对应的路由配置path:'/home/:id'或者path:'home:id'，否则刷新浏览器参数丢失

3.直接通过url传参，push({path: '/home/123'})或者push('/home/123')，对应的路由配置path:'/home/:id'，刷新浏览器参数不会丢失

<https://www.jianshu.com/p/ea0e39a8401b>

router方法的封装：

把一个公共路由方法封装成一个方法然后引用该方法即可：

把 this.$router.push方法封装成方法然后引用：

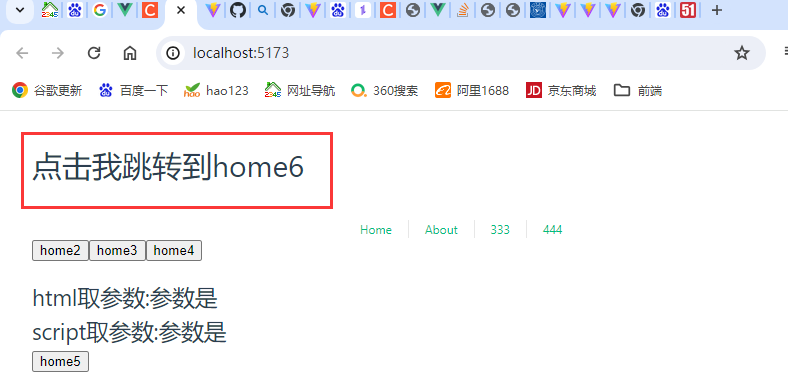
在组件中：info代表的是路由的参数



引用：然后写入参数（路径和参数）



页面中显示：



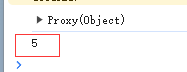
点击之后跳转到home4页面：

证明这个方法的引用没有问题。Hom4页面写入了获取参数的代码：





路径的参数是5，查看控制台我们可以看到id的打印：

 获取成功