

Vite4 + Vue3 + vue-router4 动态路由

Etc.End 已于 2023-07-13 10:19:27 修改
于 2023-03-12 12:47:29 首次发布

动态路由，基本上每一个项目都能接触到这个东西，通俗一点就是我们的菜单是根据后端接口返回的数据进行动态生成的。表面上是对菜单的一个展现处理，其实内部就是对router的一个数据处理。当然你只对菜单做处理也是可以的，但是没有任何意义，熟悉router的小伙伴都知道，如果你的一个路由存在，即使没有这个菜单，我只要改变浏览器的地址一样能访问到。所以你还是省不了修改router的步骤。通过接口获取数据的话就可以根据角色权限或者一些业务上的需求，根据不同属性实现路由的划分。达到不同的页面渲染效果。

本文只是讲解菜单的 **权限控制**，不到按钮级别。其实按钮也是差不多的。可以设置一个属性表示菜单，一个属性表示按钮，每一个菜单的叶子节点上都包含根据权限返回的按钮数组。接着可以通过组件的形式去输出相应的按钮就可以。

一、搭建项目 🤖🤖🤖

这里我已经提前搭建好了 Vite4+Pinia2+vue-router4+ElementPlus搭建Vue3项目(组件、图标等按需引入)

二、根据上面链接搭建好项目，修改src/router/index.ts 😊😊

asyncRoutes里面可以存放默认的一些路由，比如登录、404、403这些。由于我只是演示，所以就啥都不放了。清晰明朗一点。

```
1 import { createRouter, createWebHashHistory, RouteRecordRaw } from 'vue-router'
2
3 export const asyncRoutes: RouteRecordRaw[] = []
4
5 const router = createRouter({
6   history: createWebHashHistory(),
7   routes: asyncRoutes,
8   scrollBehavior: () => ({ left: 0, top: 0 })
9 })
10
11 export default router
```

三、创建 'src/layout/index.vue' 文件 😊😊

这个文件就是整个项目的布局，一般我们常见的项目都分为上下结构，如下图。导航和菜单部分基本上是用户登录以后就已经确定好了，点击菜单的时候去切换路由。我这里由于这部分不是重点，所以我就很潦草的画了一个很简单的页面。



CSDN @Etc.End

```
1 <template>
2   <div style="padding: 100px;">
3     <div>
4       <div v-for="(item, index) in menus[0].children" :key="index" style="margin-bottom: 20px;">
5         <router-link :to="item.path">{{item.title}}</router-link>
6       </div>
7     </div>
8     <div>
9       <router-view #default="{route, Component}">
10        <transition leave-from-class="ts-web-fade--leave-to" enter-active-class="animate__animated animate__bounceInRight">
11          <component :is="Component"></component>
12        </transition>
13      </router-view>
14    </div>
15  </div>
16 </template>
```

```
13 |         </router-view>14 |         </div>
15 |     </div>
16 | </template>
17 |
18 |
19 | <script lang="ts">
20 | import appStore from '@pinia';
21 |
22 | export default defineComponent({
23 |   setup() {
24 |
25 |     const { menus } = appStore.permissionModule
26 |
27 |     console.log(menus, 'menus')
28 |
29 |     return {
30 |       menus,
31 |     }
32 |   }
33 | })
34 | </script>
```

四、创建 'src/pinia/modules/permission.ts' 文件 😊 😊

由于我这里是一个demo，没有真正的去接入后端。所以我暂时放入的静态数据。自己替换成接口返回就好。

```
1 | import { defineStore } from 'pinia';
2 | import router from '@router'
3 | // 这是整个项目的布局页面。根据自己的项目替换就好
4 | import Layout from "@layout/index.vue";
5 | import {RouteRecordRaw} from "vue-router";
6 |
7 | export type MenuType = {
8 |   path: string,
9 |   title: string,
10 |   component: string,
11 |   redirect?: string,
12 |   children?: Array<MenuType>
13 | }
14 |
15 | type RouterType = RouteRecordRaw & {
16 |   hidden?: boolean;
17 |   alwaysShow?: boolean;
18 | }
19 |
20 | export interface IPermissionState {
21 |   routes: RouterType[]
22 |   dynamicRoutes: RouterType[]
23 |   menus: Array<MenuType>
24 | }
25 |
26 | function hasPermission<T>(roles: T[], route: RouterType) {
27 |   if (route.meta && route.meta.roles) {
28 |     return roles.some((role) => (route.meta?.roles as T[]).includes(role));
29 |   }
30 |   return true
31 | }
32 |
33 | const modules = import.meta.glob('../views/**/*.vue')
34 | const _import = (path: string) => () => import(`../views/${path}.vue`)
35 |
36 | const assembleRouter = (routers: any) => {
37 |   const addRouter = routers.filter((router: any) => {
38 |     (router.title && router.icon) && (router.meta = {
39 |       title: router.title,
40 |       // icon: router.icon,
41 |       // alwaysShow: router.alwaysShow || false,
42 |       // affix: router.affix || false,
43 |     })
```

```

44 |         if (router.component === 'Layout') { 45 |             router.component = shallowRef(Layout)
46 |         } else {
47 |             if (import.meta.env.MODE === 'development') {
48 |                 router.component = _import(router.component)
49 |             } else {
50 |                 router.component = modules[`${../../views/${router.component}.vue`}
51 |             ]
52 |         }
53 |         if (router.children && router.children.length) {
54 |             router.children = assembleRouter(router.children)
55 |         }
56 |         return true
57 |     })
58 |     return addRouter
59 | }
60 |
61 |
62 | export function filterAsyncRoutes(routes: RouterType[], roles: string[]) {
63 |     const res: RouterType[] = []
64 |
65 |     routes.forEach((route) => {
66 |         const tmp = { ...route }
67 |         if (hasPermission<string>(roles, tmp)) {
68 |             if (tmp.children) {
69 |                 tmp.children = filterAsyncRoutes(tmp.children, roles)
70 |             }
71 |             res.push(tmp)
72 |         }
73 |     })
74 |     return res
75 | }
76 |
77 | export const permissionModule = defineStore({
78 |     id: 'permission',
79 |     state(): IPermissionState{
80 |         return {
81 |             routes: [],
82 |             dynamicRoutes: [],
83 |             menus: []
84 |         }
85 |     },
86 |     actions:{
87 |         async getMenus() {
88 |             try {
89 |                 // 这里由于不方便演示，所以我写的静态数据。换着自己对于的接口就好
90 |                 const list:MenuType[] = [
91 |                     {
92 |                         path: '/',
93 |                         title: 'ts-super-web',
94 |                         component: 'Layout',
95 |                         redirect: '/home',
96 |                         children: [
97 |                             {
98 |                                 title: 'home',
99 |                                 path: 'home',
100 |                                 component: 'home'
101 |                             },
102 |                             {
103 |                                 title: 'home1',
104 |                                 path: 'home1',
105 |                                 component: 'home1'
106 |                             }
107 |                         ]
108 |                     }
109 |                 ]
110 |                 this.menus = list
111 |                 // 组件路由
112 |                 let addRouter = assembleRouter(this.menus)
113 |                 // addRouter = assembleRouterDelete(addRouter)
114 |                 // 动态添加菜单

```

```
115 |             addRouter.forEach((ts: any) => {116 |                 router.addRoute(ts)
117 |             })
118 |         } catch (err) {
119 |             return Promise.reject(err);
120 |         }
121 |     }
122 | }
123 | })
```

五、路由拦截器 😊 😊 😊

因为上面说到我没有真正的接入后端，所以这里我也没有进行token判断。自行增加一下就好，比较简单。除了token还可以在拦截器里面放置一个白名单列表，对于白名单里面的路由我们不做拦截。比如login登录页面、404等等。根据自己需求配置就好。

```
1 | import router from '@router'
2 | // @ts-ignore
3 | import NProgress from 'nprogress'
4 | import 'nprogress/nprogress.css'
5 | import appStore from "@pinia";
6 |
7 | NProgress.configure({
8 |     easing: 'ease', // 动画方式
9 |     showSpinner: true, // 是否显示加载ico
10 |     trickleSpeed: 200, // 自动递增间隔
11 |     minimum: 0.4, // 更改启动时使用的最小百分比
12 | })
13 |
14 | router.beforeEach(async (to, form, next) => {
15 |     // 这里处理自己的逻辑,比如需要登录以后才能访问其他页面等等
16 |     NProgress.start()
17 |     const { menus, getMenus } = appStore.permissionModule
18 |     if (menus.length === 0) {
19 |         try {
20 |             // 调用接口获取菜单 进行跳转
21 |             await getMenus()
22 |             next({ ...to, replace: true })
23 |         } catch (err) {
24 |             next()
25 |         }
26 |     } else {
27 |         next()
28 |     }
29 |     NProgress.done()
30 | })
```

六、修改app.vue

```
1 | <template>
2 |   <router-view />
3 | </template>
4 |
5 | <style>
6 | html,body,#app {
7 |   height: 100%;
8 |   width: 100%;
9 |   margin: 0;
10 |   padding: 0;
11 | }
12 | </style>
13 |
```

七、src/views下的两个home文件进行一下修改

home 这两个文件不修改也不影响。

```
1 | <template>
2 |   <p style="font-size: 32px;">你好，我是home</p>
```

home1 这两个文件不修改也不影响。

```
1 <template>
2   <p style="font-size: 32px;">你好，我是home1</p>
3 </template>
4
```

八、效果浏览



我是Etc.End。如果文章对你有帮助，能否帮我点个免费的赞和收藏👍。



CSDN @Etc.End



文章知识点与官方知识档案匹配，可进一步学习相关知识

Vue入门技能树 vue3基础（JS） 构建工具Vite 38105 人正在系统学习中

