# 7.0 前端和后端的结合

一、pinia引入

//"先注册 -> 再配置 -> 最后挂载" 的生命周期顺序  
import { createApp } from 'vue';  
import App from './App.vue';  
import "@/assets/less/index.less";  
import ElementPlus from 'element-plus';  
import 'element-plus/dist/index.css';  
import \* as ElementPlusIconsVue from '@element-plus/icons-vue';  
import router from './router';  
import { createPinia } from 'pinia';  
import api from "@/api/api";  
import {useAllDataStore} from "@/stores"  
  
 //getRoutes获得所有路由记录的完整列表。  
 //这个方法判断要跳转的路由是否存在  
 function isRoute(to){  
 return router.getRoutes().filter(item=>item.path===to.path).length>0  
}  
  
router.beforeEach((to, from) => {  
 //如果要跳转的不是login,且token不存在(可以通过不存在token判断出用户未登录)  
 if(to.path !== '/login'&&!store.state.token){  
 //跳转到login  
 return { name: 'login' }  
 }  
 //如果路由记录不存在  
 if(!isRoute(to)){  
 //跳转到404界面  
 return {name: "404"}  
 }  
})  
  
const app = createApp(App);  
const pinia = createPinia();  
  
  
// 注册图标组件 (必须在挂载前完成)  
for (const [key, component] of Object.entries(ElementPlusIconsVue)) {  
 app.component(key, component);  
}  
  
// 按顺序使用插件  
  
app.use(pinia);  
app.use(ElementPlus);  
//user pinia 之后 use router之前  
const store=useAllDataStore();  
store.addMenu(router,"refresh")  
  
app.use(router);  
// 配置全局属性  
app.config.globalProperties.$api = api;  
//按需加载 mock  
// if (import.meta.env.DEV) {  
// import("@/api/mock.js");  
// }  
import("@/api/mock.js");  
  
// 最后挂载  
app.mount('#app');

二、src目录下子目录stores下index.js的内容

//"先注册 -> 再配置 -> 最后挂载" 的生命周期顺序  
import { createApp } from 'vue';  
import App from './App.vue';  
import "@/assets/less/index.less";  
import ElementPlus from 'element-plus';  
import 'element-plus/dist/index.css';  
import \* as ElementPlusIconsVue from '@element-plus/icons-vue';  
import router from './router';  
import { createPinia } from 'pinia';  
import api from "@/api/api";  
import {useAllDataStore} from "@/stores"  
  
 //getRoutes获得所有路由记录的完整列表。  
 //这个方法判断要跳转的路由是否存在  
 function isRoute(to){  
 return router.getRoutes().filter(item=>item.path===to.path).length>0  
}  
  
router.beforeEach((to, from) => {  
 //如果要跳转的不是login,且token不存在(可以通过不存在token判断出用户未登录)  
 if(to.path !== '/login'&&!store.state.token){  
 //跳转到login  
 return { name: 'login' }  
 }  
 //如果路由记录不存在  
 if(!isRoute(to)){  
 //跳转到404界面  
 return {name: "404"}  
 }  
})  
  
const app = createApp(App);  
const pinia = createPinia();  
  
  
// 注册图标组件 (必须在挂载前完成)  
for (const [key, component] of Object.entries(ElementPlusIconsVue)) {  
 app.component(key, component);  
}  
  
// 按顺序使用插件  
  
app.use(pinia);  
app.use(ElementPlus);  
//user pinia 之后 use router之前  
const store=useAllDataStore();  
store.addMenu(router,"refresh")  
  
app.use(router);  
// 配置全局属性  
app.config.globalProperties.$api = api;  
//按需加载 mock  
// if (import.meta.env.DEV) {  
// import("@/api/mock.js");  
// }  
import("@/api/mock.js");  
  
// 最后挂载  
app.mount('#app');

三、src目录下子目录views目录下Login.vue的内容

<template>  
 <div class="body-login">  
 <el-form :model="loginForm" class="login-container">  
 <h3>欢迎登录</h3>  
 <el-form-item>  
 <el-input type="input" placeholder="请输入账户" v-model="loginForm.username"></el-input>  
 </el-form-item>  
 <el-form-item>  
 <el-input type="password" placeholder="请输入密码" v-model="loginForm.password"></el-input>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" @click="login"> 登录 </el-button>  
 </el-form-item>  
 </el-form>  
 </div>  
</template>  
<script setup>  
  
import {reactive,getCurrentInstance} from 'vue'  
import { useAllDataStore } from '@/stores/index.js'  
import { useRouter } from 'vue-router'  
  
const { proxy } = getCurrentInstance()  
const store=useAllDataStore()  
const router=useRouter()  
const loginForm = reactive({  
 username: 'admin',  
 password: 'admin',  
});  
  
const login=async ()=>{  
 const res = await proxy.$api.getMenu(loginForm);  
 if(res){  
 store.updateMenuList(res.menuList)   
 //在这里执行添加路由方法,并传入router  
 store.addMenu(router)   
 store.state.token=res.token   
 console.log(res)  
 router.push("/home")  
 }  
}  
</script>  
<style scoped lang="less">  
.body-login{   
 position: fixed;  
 top: 0;  
 left: 0;  
 width: 100vw;  
 height: 100vh;  
 background-image: url("../assets/images/background.png");  
 background-size: cover;  
 background-position: center;  
 background-repeat: no-repeat;  
 display: flex;  
 align-items: center;  
 justify-content: center;  
}  
.login-container {  
 width: 350px;  
 background-color: #fff;  
 border: 1px solid #eaeaea;  
 border-radius: 15px;  
 padding: 35px 35px 15px 35px;  
 box-shadow: 0 0 25px #cacaca;  
 margin: 180px auto;  
 h3 {  
 text-align: center;  
 margin-bottom: 20px;  
 color: #505450;  
 }  
 :deep(.el-form-item\_\_content) {  
 justify-content: center;  
 }  
}  
</style>