0. 实现步骤

1. 安装并配置 vue-router 4.x
2. 展示 Login.vue 登录组件
3. 模拟并实现登录功能
4. 通过路由渲染 Home.vue
5. 实现退出登录的功能
6. 全局控制路由的访问权限
7. 将左侧菜单改造为路由链接
8. 渲染用户管理页面的数据
9. 实现跳转到用户详情页的功能
10. 开启路由的 props 传参
11. 通过编程式导航实现后退功能

1. 安装并配置 vue-router 4.x

1. 运行如下的命令，安装 `vue-router`：

```
npm install vue-router@next -S
```

2. 在 `src` 目录下新建 `router.js` 路由模块：

```
// 1. 按需导入对应的函数
import { createRouter, createWebHashHistory } from 'vue-router'

// 2. 创建路由对象
const router = createRouter({
  history: createWebHashHistory(),
  routes: [],
})

// 3. 向外共享路由实例对象
export default router
```

3. 在 `main.js` 入口文件中导入并挂载路由对象：

黑马程序员 www.itheima.com

```
// 1. 导入路由模块
import router from './router'

const app = createApp(App)

// 2. 挂载路由对象
app.use(router)

app.mount('#app')
```

2. 展示 Login.vue 登录组件

1. 在 `router.js` 模块中导入 Login.vue 组件：

```
import Login from './components/MyLogin.vue'
```

2. 声明路由规则如下：

```
routes: [
    // 路由重定向
    { path: '/', redirect: '/login' },
    { path: '/login', component: Login },
]
```

3. 在 `App.vue` 组件中声明**路由占位符**：

```
<template>
  <!-- 路由的占位符 -->
  <router-view></router-view>
</template>

<script>
export default {
  name: 'MyApp',
}
</script>

<style lang="less" scoped>
</style>
```

黑马程序员 www.itheima.com

3. 模拟并实现登录功能

1. 在 `MyLogin.vue` 组件中声明如下的 `data` 数据：

```
data() {
    return {
        username: '',
        password: '',
    }
},
```

2. 为**用户名**和**密码**的文本框进行 `v-model` 双向数据绑定：

```
<!-- 登录名称 -->
<div class="form-group form-inline">
  <label for="username">登录名称</label>
  <input type="text" class="form-control ml-2" id="username" placeholder="请输入登录名称" autocomplete="off" v-model="username">
</div>

<!-- 登录密码 -->
<div class="form-group form-inline">
  <label for="password">登录密码</label>
  <input type="password" class="form-control ml-2" id="password" placeholder="请输入登录密码" v-model="password">
</div>
```

3. 为 **登录按钮** 绑定点击事件处理函数：

```
<button type="button" class="btn btn-primary" @click="onLoginClick">登录</button>
```

4. 在 `methods` 中声明 `onLoginClick` 事件处理函数如下：

黑马程序员 www.itheima.com

```
methods: {
onLoginClick() {
  // 判断用户名和密码是否正确
  if (this.username === 'admin' && this.password === '123456') {
    // 登录成功，跳转到后台主页
    this.$router.push('/home')
    // 模拟存储 Token 的操作
    return localStorage.setItem('token', 'Bearer xxx')
  }
  // 登录失败，清除 Token
  localStorage.removeItem('token')
},
},
```

4. 通过路由渲染 Home.vue

1. 在 `router.js` 中导入 `Home.vue` 组件：

```
import Home from './components/MyHome.vue'
```

2. 在 `routes` 路由规则的数组中，声明对应的路由规则：

```
routes: [
    { path: '/', redirect: '/login' },
    { path: '/login', component: Login },
    // Home 组件的路由规则
    { path: '/home', component: Home },
]
```

3. 渲染 `Home.vue` 组件的基本结构：

```
<template>
  <div class="home-container">
    <!-- 头部组件 -->
    <my-header></my-header>

    <!-- 主体区域 -->
    <div class="home-main-box">

```

黑马程序员 www.itheima.com

```
        <!-- 左侧边栏区域 -->
        <my-aside></my-aside>

        <!-- 右侧内容主体区域 -->
        <div class="home-main-body"></div>

      </div>
    </div>
  </template>
```

5. 实现退出登录的功能

1. 在 `MyHeader.vue` 组件中，为 `退出登录` 按钮绑定 `click` 事件处理函数：

```
<button type="button" class="btn btn-light" @click="onLogout">退出登录</button>
```

2. 在 `methods` 中声明如下的事件处理函数：

```
export default {
  name: 'MyHeader',
  methods: {
    // 退出按钮的点击事件处理函数
    onLogout() {
      // 移除 Token
      localStorage.removeItem('token')
      // 强制跳转到“登录页面”
      this.$router.push('/login')
    },
  },
}
```

6. 全局控制路由的访问权限

1. 在 `router.js` 模块中，通过 `router` 路由实例对象，全局挂载路由导航守卫：

黑马程序员 www.itheima.com

```js
// 全局路由导航守卫
router.beforeEach((to, from, next) => {
  // 如果用户访问的是登录页面，直接放行
  if (to.path === '/login') return next()
  // 获取 Token 值
  const token = localStorage.getItem('token')
  if (!token) {
    // Token 值不存在，强制跳转到登录页面
    return next('/login')
  }
  // 存在 Token 值，直接放行
  next()
})
```

7. 将左侧菜单改造为路由链接

1. 打开 `MyAside.vue` 组件，把 `li` 内部的纯文本升级改造为 `<router-link>` 组件：

```vue
<template>
  <div class="layout-aside-container">
    <!-- 左侧边栏列表 -->
    <ul class="user-select-none menu">
      <li class="menu-item">
        <router-link to="/home/users">用户管理</router-link>
      </li>
      <li class="menu-item">
        <router-link to="/home/rights">权限管理</router-link>
      </li>
      <li class="menu-item">
        <router-link to="/home/goods">商品管理</router-link>
      </li>
      <li class="menu-item">
        <router-link to="/home/orders">订单管理</router-link>
      </li>
      <li class="menu-item">
        <router-link to="/home/settings">系统设置</router-link>
      </li>
    </ul>
  </div>
</template>
```

黑马程序员 www.itheima.com

2. 打开 `Home.vue` 组件，在 **右侧内容主体区域** 中声明子路由的占位符：

```vue
<template>
  <div class="home-container">
    <!-- 头部组件 -->
    <my-header></my-header>
    <!-- 主体区域 -->
    <div class="home-main-box">
      <!-- 左侧边栏区域 -->
      <my-aside></my-aside>
      <!-- 右侧内容主体区域 -->
      <div class="home-main-body">
        <!-- **子路由的占位符** -->
        <router-view></router-view>
      </div>
    </div>
  </div>
</template>
```

3. 在 `router.js` 中导入左侧菜单对应的组件：

```js
import Users from './components/menus/MyUsers.vue'
import Rights from './components/menus/MyRights.vue'
import Goods from './components/menus/MyGoods.vue'
import Orders from './components/menus/MyOrders.vue'
import Settings from './components/menus/MySettings.vue'
```

4. 通过 `children` 属性，为 `home` 规则定义子路由规则如下：

```js
{
  path: '/home',
  component: Home,
  // 用户访问 /home 时，重定向到 /home/users
  redirect: '/home/users',
  // 子路由规则
  children: [
    { path: 'users', component: Users },
    { path: 'rights', component: Rights },
    { path: 'goods', component: Goods },
    { path: 'orders', component: Orders },
    { path: 'settings', component: Settings },
  ],
},
```

黑马程序员 www.itheima.com

8. 渲染用户管理页面的数据

1. 在 `MyUsers.vue` 组件中，通过 `v-for` 指令循环渲染用户列表的数据：

```
<tbody>
  <tr v-for="(item, i) in userlist" :key="item.id">
    <td>{{i + 1}}</td>
    <td>{{item.name}}</td>
    <td>{{item.age}}</td>
    <td>{{item.position}}</td>
    <td>详情</td>
  </tr>
</tbody>
```

9. 实现跳转到用户详情页的功能

1. 在 `MyUsers.vue` 组件中，渲染详情页的**路由链接**如下：

```
<td>
  <router-link :to="'/home/users/' + item.id">详情</router-link>
</td>
```

2. 在 `router.js` 中导入**用户详情页**组件：

```
import UserDetail from './components/user/MyUserDetail.vue'
```

3. 在 `home` 规则的 `children` 节点下，声明 **用户详情页** 的路由规则：

```
{
  path: '/home',
  component: Home,
  redirect: '/home/users',
  children: [
    { path: 'users', component: Users },
    { path: 'rights', component: Rights },
    { path: 'goods', component: Goods },
    { path: 'orders', component: Orders },
```

黑马程序员 www.itheima.com

```
      { path: 'settings', component: Settings },
      // 用户详情页的路由规则
      { path: 'users/:id', component: UserDetail },
    ],
  },
```

10. 开启路由的 props 传参

1. 在 `router.js` 模块中，为 **用户详情页** 的路由规则开启 `props` 传参：

```
{ path: 'users/:id', component: UserDetail, props: true },
```

2. 在 `MyUserDetail.vue` 组件中声明 `props` 参数：

```
export default {
  name: 'MyUserDetail',
  props: ['id'],
}
```

3. 在 `MyUserDetail.vue` 组件的结构中直接使用路由参数：

```
<template>
  <button type="button" class="btn btn-light btn-sm">后退</button>
  <h4 class="text-center">用户详情 --- {{id}}</h4>
</template>
```

11. 通过编程式导航实现后退功能

1. 在 `MyUserDetail.vue` 组件中，为**后退按钮**绑定点击事件处理函数：

```
<template>
  <button type="button" class="btn btn-light btn-sm"
@click="goBack">后退</button>
  <h4 class="text-center">用户详情 --- {{id}}</h4>
</template>
```

2. 在 `methods` 中声明 `goBack` 事件处理函数如下：

黑马程序员 www.itheima.com

```
export default {
  name: 'MyUserDetail',
  props: ['id'],
  methods: {
    // 编程式导航实现后退功能
    goBack() {
      this.$router.go(-1)
    },
  },
}
```

