

04.Router 入门和安装

学习要点：

1. 安装 router
2. 路由详解

本节课我们来开始了解 Vue router 的入门和安装方法。

一. 安装 router

1. Vue router 是 Vue.js 的官方路由管理器，用于构建单页面项目；
2. 在安装 Vue-cli 时，选择 router 即可安装，期间有一个是否启用 history 选项；
3. 直接选择 yes 启用即可，如果想测试 no(hash 模式)，可修改配置更改；
4. 安装好后，我们 npm run serve 启动项目，会发现多了两个 link，单页链接；
5. 区别：history 模式则没有这个#号，而 hash 模式会出现#号，前者需要后端支持；
6. 具体更改模式在：src/router/index.js 里，设置 options；

//history 模式

```
import { createRouter, createWebHistory } from 'vue-router'

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})
```

//hash 模式

```
import { createRouter, createWebHashHistory } from 'vue-router'

const router = createRouter({
  history: createWebHashHistory(),
  routes
})
```

PS：设置 history 模式，需要服务器环境配置的支持，手册已经给出了各个服务的解决方案；

PS：地址为：<https://router.vuejs.org/zh/guide/essentials/history-mode.html>；

二. 路由详解

1. 安装了 router 路由的脚手架，多了两个目录，且配置和入口文件也改变了；
2. 先来分析下入口文件 main.js，比之前多了引入和注册路由的内容；

// 引入路由

```
import router from './router'
```

// 注册路由

```
createApp(App).use(router).mount('#app')
```

PS：.use()可以参考 API 手册，意味：注册一个插件；

<https://vue3js.cn/docs/zh/api/application-api.html#use>

3. 我们来分析一下 App.vue 中路由的各种元素，搭配 router/index.js 对照看；

```
<!--<router-link>标签-->
<router-link to="/">Home</router-link> |
<router-link to="/about">About</router-link>
```

PS: 分别说明含义:

- (1) .<router-link>是组件导航，会被渲染成<a>超链接标签；
- (2) .to 属性是链接地址，会被渲染成 href 属性；

4. <router-link>对于的 index.js 部分，我们了解下：

```
// 引入 vue-router , 并导入两个方法
// createRouter 用于创建路由对象
// createWebHistory 用于启用无#号的 history 模式
import { createRouter, createWebHistory } from 'vue-router'
// 引入 Home 组件，就是起始页面
import Home from '../views/Home.vue'

// 数组用于设置每个组件的信息
const routes = [
  {
    path: '/',
    name: 'Home',
    component: Home
  },
  ...
]

// 路由对象，实例化
const router = createRouter({
  // 开启 history 模式，去#号
  history: createWebHistory(process.env.BASE_URL),
  // ES6 语法，key 和 value 相同时可以简写
  // routes : routes
  routes
})

//导出路由
export default router
```

5. 再回到 App.vue，导航链接下方有一个空标签<router-view/>，用于渲染页面；
6. <router-view/>渲染 views 目录下的页面组件，而 components 是功能性小组件；