# Vue day7

## 路由

1. 什么是路由

路由（router）就是对应关系

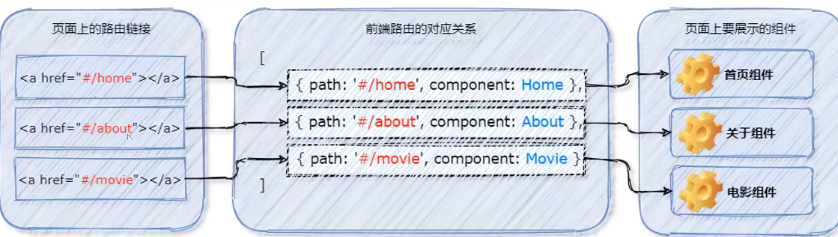
1. spa与前端路由

spa指的是一个web网站只有唯一的一个HTML页面，所有组件的展示与切换都在这唯一的一个页面完成。此时，不同组件之间的切换需要通过前端路由来实现

1. 什么是前端路由

通俗易懂的概念：Hash地址与组件之间的对应关系

1. 前端路由的工作方式
2. 用户点击了页面上的路由链接
3. 导致了URL地址中的Hash值发生了变化
4. 前端路由监听到了Hash值变化
5. 前端路由把当前Hash地址对应的组件渲染到浏览器中



1. 实现简易前端路由
2. 通过<component>标签，结合comName动态渲染组件。示例：



1. 通过在created生命周期中，调用window.onhashchange事件来监听hash地址的变动，通过location.hash拿到hash地址，同时修改comName的值，渲染不同的组件



1. vue-router
   1. 什么是vue-router

vue-router是vue.js官方给出的路由解决方案。它只能结合vue项目进行使用，能够轻松的管理SPA项目中的组件切换。

vue-router的官方中文地址：<https://router.vuejs.org/zh/>

* 1. vue-router安装和配置的步骤

1. 安装vue-router包
2. 创建路由模块
3. 导入并挂载路由模块
4. 声明路由链接和占位符

6.2.1 在项目中安装vue-router

在vue2的项目中，安装vue-router的命令如下：

npm i [vue-router@3.5.2](mailto:vue-router@3.5.2) -S

6.2.2 创建路由模块

在src源代码目录下，新建router/index.js路由模块，并初始化，示例如下：

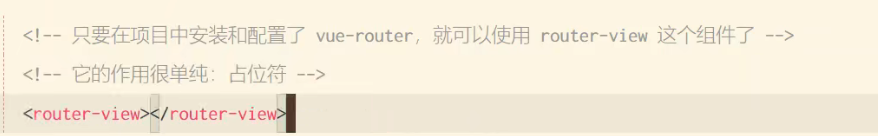


然后再main.js中挂载路由：



6.2.3在路由模块中声明路由的对应关系

（1） 当配置好了vue-router之后就可以在组件中使用<router-view></router-view>组件来调用路由组件，例：在app.vue组件中：



（2）在路由模块中的实例对象上，配置对应关系

这里是index.js中：



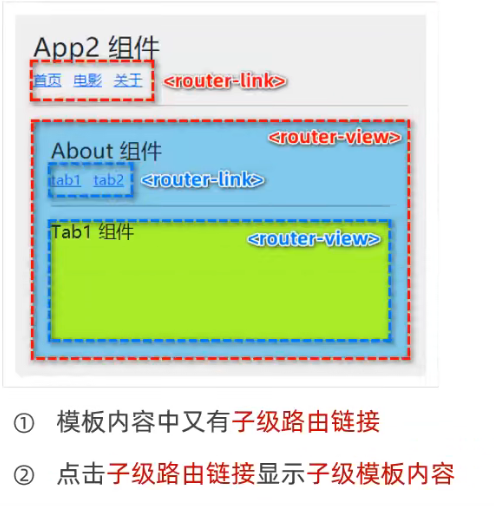
* 1. 路由重定向

路由重定向指的是：用户在访问地址A的时候，强制用户跳转到地址C，从而展示特定的组件页面。通过路由规则的redirect属性，指定一个新的路由地址，可以方便地设置路由的重定向：



1. 嵌套路由

通过路由实现组件的嵌套展示，叫做嵌套路由





* 1. 通过children属性声明子路由规则

在案例中，在src/router/index.js路由模板中，导入需要的组件，并使用children属性声明子路由规则：



* 1. 子路由重定向

在定义子路由规则的时候也可以添加重定向规则：



* 1. 默认子路由

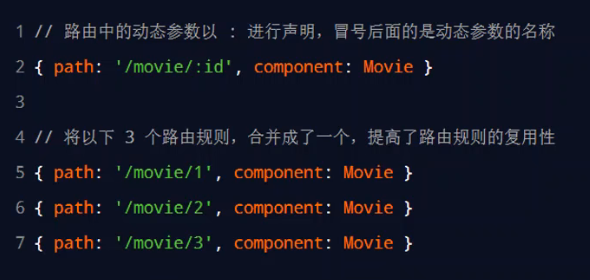
如果children数组中，某个路由规则的path值为空字符串，则这条路由规则叫做“默认子路由”

1. 动态路由匹配

8.1动态路由的概念

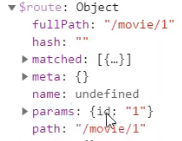
动态路由指的是：把hash地址中可变的部分定义为参数项，从而提高路由规则的复用性。

在vue-router中使用英文的冒号（:）定义路由的参数项。示例：



获取参数方法1：

在路由访问要渲染的组件中，它的实例对象（this）下有$route对象，$route对象下又有params对象，在params对象中就存放着我们自定义的动态参数的值，所以可以通过this.$route.params.xxx来获得属性值，然后根据属性值渲染对应的内容



注：$route是路由的“参数对象”，$router是路由的“导航对象”

获取参数方法2：

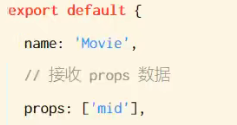
通过props获取参数

在路由规则中展示的组件后添加 props：ture即可

例：

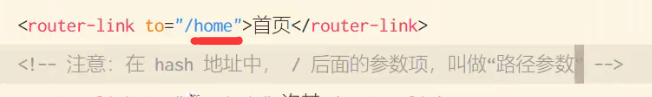


在组件中定义props自定义属性，用于接收



注意项：

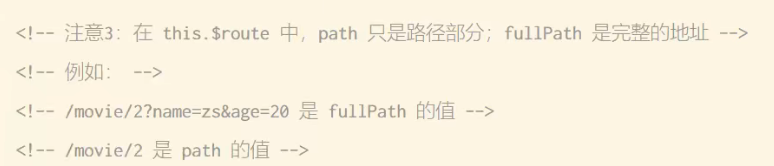
1.在hash地址中 /后面的参数项，叫做“路径参数”



2.在hash地址中 ?后面的参数项，叫“查询参数”



3.



1. 导航 ：声明式导航和编程式导航

声明式导航：

在浏览器中，点击链接实现导航的方式，叫做声明式导航。例如：

••普通网页中点击<a>链接、vue项目中点击<router-link>都属于声明式导航

编程式导航：

在浏览器中，调用API方法实现导航的方式，叫做编程式导航。例如：

•普通网页中调用location.href跳转到新页面的方式，属于编程式导航

* 1. vue-router中的编程式导航API

vue-router提供了许多编程式导航的API，其中最常用的导航API分别是：

1. this.$router.push(‘hash地址’)

跳转到指定的hash地址，并增加一条历史记录

1. this.$router.replace(‘hash地址’)

跳转到指定的hash地址，并替换掉当前的历史记录

1. this.$router.go(数值n)

调用这个方法，可以在浏览历史中前进和后退（n为负数代表后退为正就前进，数值表示前进或后退多少个历史记录，超过上限则保持不动）。示例：



在实际开发中，一般只前进或后退一层页面。因此vue-router提供了如下两个便捷方法：

【1】$router.back()

后退一个页面

【2】$router.forward()

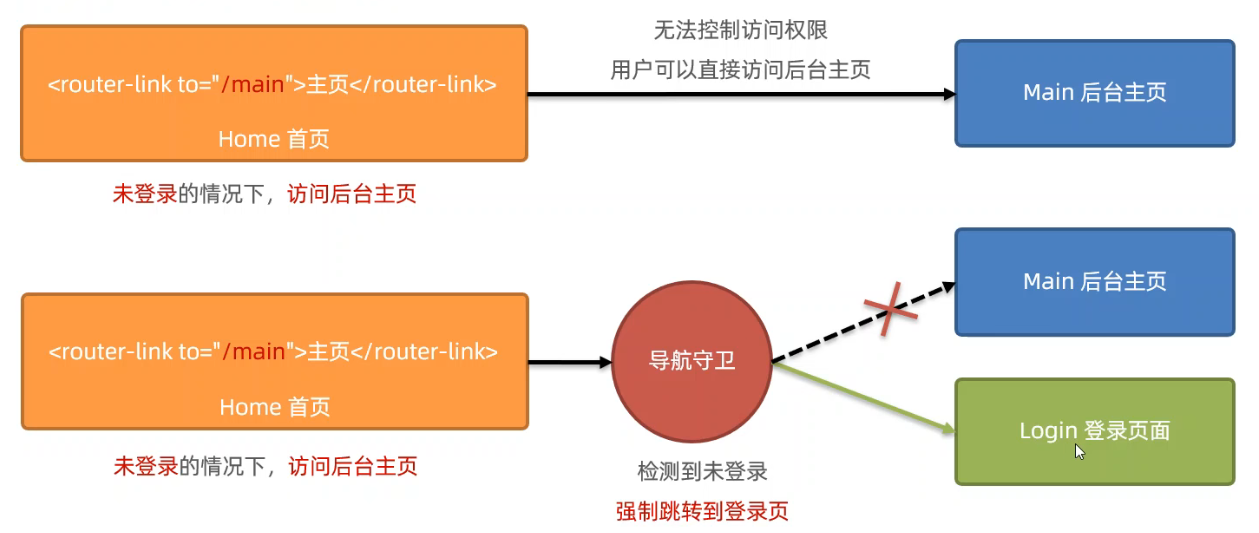
前进一个页面

注意：



* 1. 导航守卫

导航守卫可以控制路由的访问权限。示意图如下：



* + 1. 全局前置守卫

每次发生路由的导航跳转时，都会触发全局前置守卫。因此，在全局前置守卫中，程序员可以对每个路由进行访问权限的控制：



守卫方法的三个形参：

全局前置守卫的回调函数中接收3个形参，格式为：



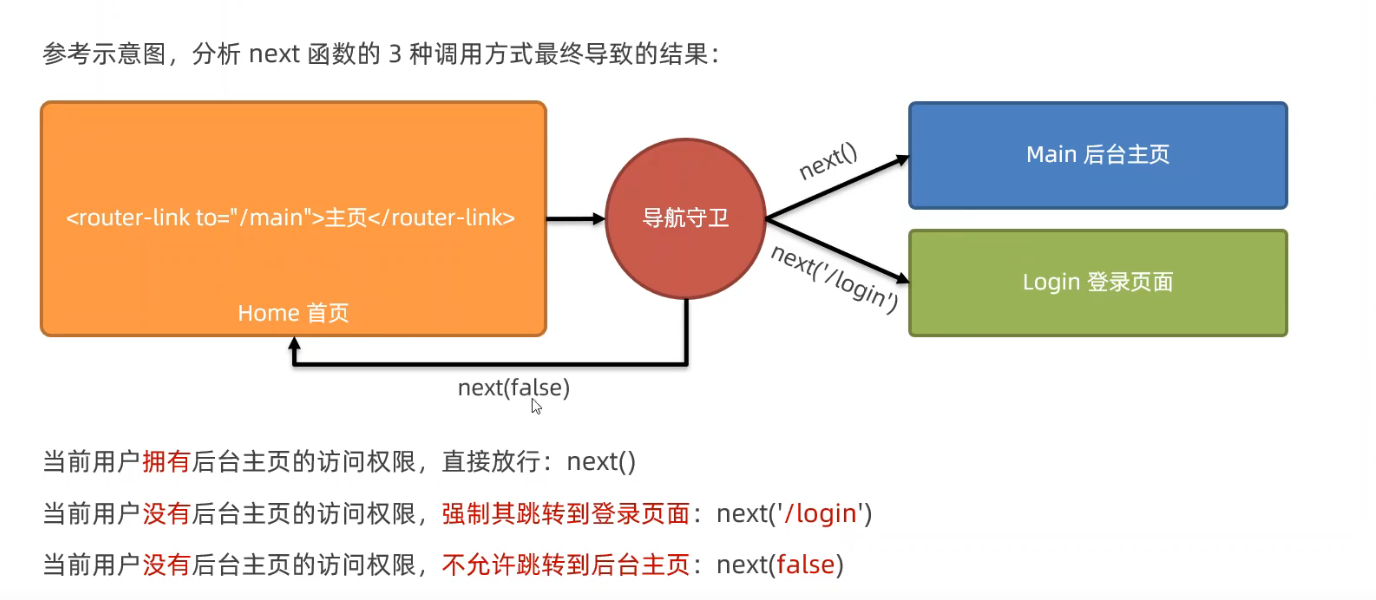
参数to：要访问的地址的信息

参数form：到下一个页面时，所在的上一个页面的信息

参数next()：放行函数

next()函数的三种调用方式：

参考示意图，分析next函数的3种调用方式最终导致的结果：



控制后台主页访问权限：

