目录

[第一节 前端路由的概念与原理 1](#_Toc27272)

[1.什么是路由 1](#_Toc10921)

[2.SPA与前端路由 1](#_Toc27929)

[3.什么是前端路由 1](#_Toc5189)

[4.前端路由的工作方式 1](#_Toc11899)

[5.实现简易的前端路由 2](#_Toc24349)

[第二节 Vue-router的基本使用 4](#_Toc1046)

[1.什么是Vue-router 4](#_Toc4733)

[2.Vue-router安装和配置的步骤 4](#_Toc18822)

[2.1 在项目中安装Vue-router 4](#_Toc12066)

[2.2 创建路由模块 4](#_Toc2520)

[2.3 导入并挂载路由模块 5](#_Toc5126)

[2.4 声明路由链接和占位符 5](#_Toc12220)

[3.声明路由的匹配规则 6](#_Toc21070)

[第三节 Vue-router的常见用法 7](#_Toc23592)

[1.路由重定向 7](#_Toc12758)

[2.嵌套路由 7](#_Toc16200)

[2.1 声明子路由链接和子路由占位符 8](#_Toc29425)

[2.2 通过children属性声明子路由规则 8](#_Toc28073)

[3.动态路由匹配 9](#_Toc32599)

[3.1 动态路由的概念 9](#_Toc15361)

[3.2 $route.params参数对象 10](#_Toc2902)

[3.3 使用props接收路由参数 11](#_Toc15882)

[4.声明式导航&编程式导航 11](#_Toc29124)

[4.1 vue-router中的编程式导航API 11](#_Toc10559)

[4.2 $router.push 12](#_Toc10309)

[4.3 $router.replace 12](#_Toc8832)

[4.4 $router.go 13](#_Toc22230)

[4.5 $router.go的简化用法 13](#_Toc30189)

[5.导航守卫 14](#_Toc13516)

[5.1 全局前置守卫 14](#_Toc29300)

[5.2 守卫方法的3个形参 15](#_Toc7841)

[5.3 next函数的3种调用方式 15](#_Toc12000)

[5.4控制后台主页的访问权限 16](#_Toc13120)

**第一节 前端路由的概念与原理**

**1.什么是路由**

路由（英文: router）就是对应关系。

**2.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内完成。此时，不同组件之间的切换需要通过前端路由来实现。

结论:在SPA项目中，不同功能之间的切换，要依赖于前端路由来完成!

**3.什么是前端路由**

通俗易懂的概念:Hash地址与组件之间的对应关系。

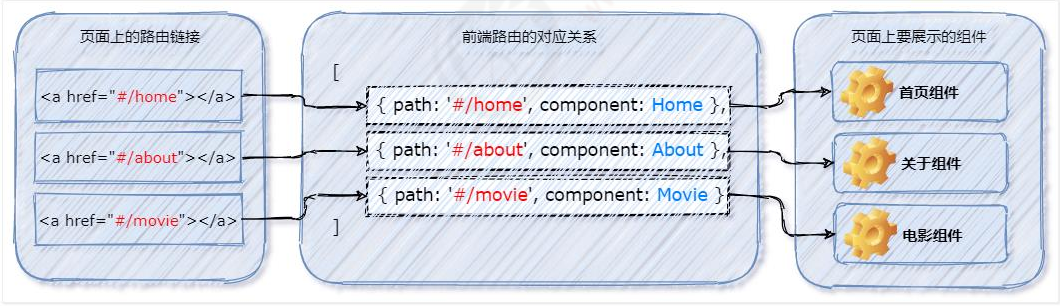
**4.前端路由的工作方式**

①用户点击了页面上的路由链接

②导致了URL地址栏中的Hash值发生了变化

③前端路由监听了到 Hash地址的变化

④前端路由把当前Hash 地址对应的组件渲染都浏览器中



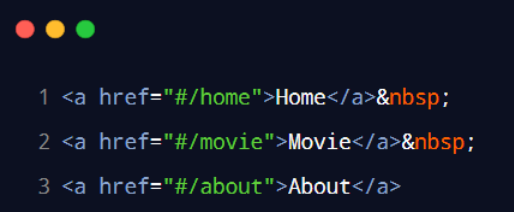
结论：前端路由，指的是Hash地址与组件之间的对应关系！

**5.实现简易的前端路由**

①步骤1:通过<component>标签，结合comName动态渲染组件。示例代码如下:



②步骤2:在App.vue组件中，为<a>链接添加对应的hash值:



③步骤3：在created生命周期函数中，监听浏览器地址栏中hash地址的变化，动态切换要展示的组件的名称:



**第二节 Vue-router的基本使用**

**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项目中组件的切换。

vue-router的官方文档地址: https: // router.vuejs.org/zh/

**2.Vue-router安装和配置的步骤**

①安装vue-router包

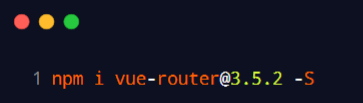
②创建路由模块

③导入并挂载路由模块

④声明路由链接和占位符

**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:



**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，并初始化如下的代码:



**2.3 导入并挂载路由模块**

在src/main.js 入口文件中，导入并挂载路由模块。示例代码如下:



在进行模块化导入的时候，如果给定的是文件夹，则默认导入这个文件夹下，名字叫做index.js 的文件

**2.4 声明路由链接和占位符**

在src/App.vue组件中，使用vue-router提供的<router-link>和<router-view>声明路由链接和占位符:



**3.声明路由的匹配规则**

在src/router/index.js路由模块中，通过routes数组声明路由的匹配规则。示例代码如下:



**第三节 Vue-router的常见用法**

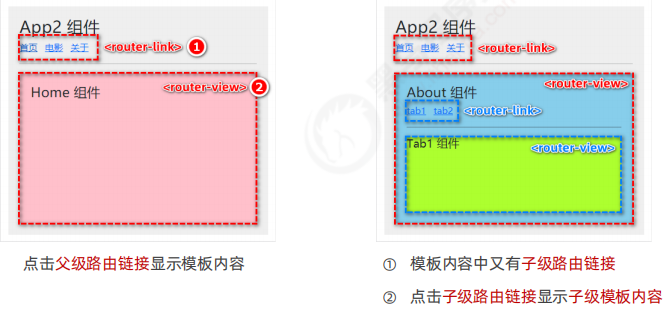
**1.路由重定向**

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，可以很方便地设置路由的重定向:



**2.嵌套路由**

通过路由实现组件的嵌套展示，叫做嵌套路由。



**2.1 声明子路由链接和子路由占位符**

在About.vue组件中，声明tab1和tab2的子路由链接以及子路由占位符。示例代码如下:

![EPS~3CR5S4Z8{JPJ](J{4@O](data:image/png;base64,)

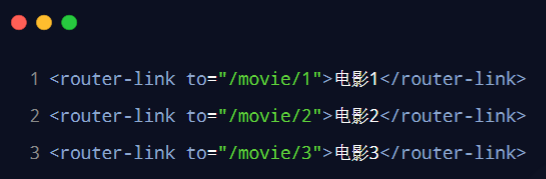
**2.2 通过children属性声明子路由规则**

在src/router /index.js路由模块中，导入需要的组件，并使用children属性声明子路由规则;只要是children里面的子路由就不用加 / 开头

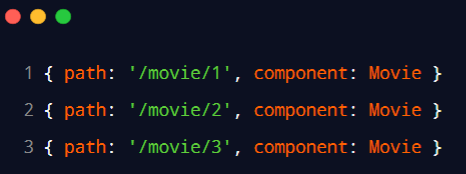
![G@A1@BD{)D`4[$N](HBHYDD](data:image/png;base64,)

**3.动态路由匹配**

思考:有如下3个路由链接:



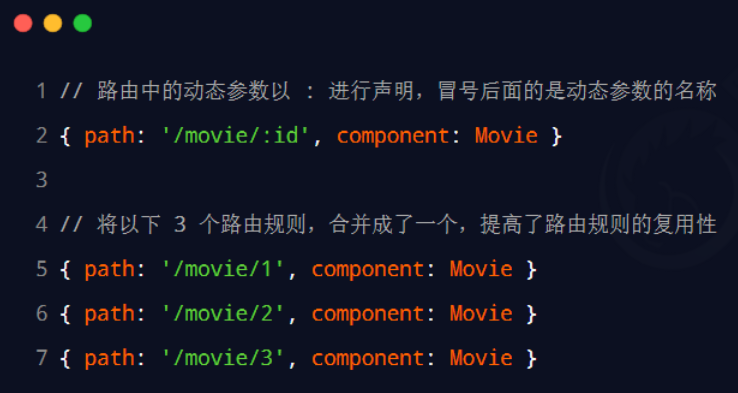
定义如下3个路由规则，是否可行???



缺点:路由规则的复用性差。

**3.1 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（:）来定义路由的**动态路径参数项**。示例代码如下:



**3.2 $route.params参数对象**

在动态路由渲染出来的组件中，可以使用this.$route.params对象访问到动态匹配的参数值。



<!--注意1:在 hash地址中, / 后面的参数项，叫做“路径参数”-->

<!--在路由“参数对象"中，需要使用this. $route.params来访问路径参数-->

<!--注意2:在 hash地址中, ? 后面的参数项，叫做“"查询参数”-->

<!--在路由“参数对象"中，需要使用this.$route . query来访问查询参数-->

<!--注意3:在this.$route中, path只是路径部分;fullPath是完整的地址-->

**3.3 使用props接收路由参数**

为了简化路由参数的获取形式，vue-router允许在路由规则中开启props传参。示例代码如下:



**4.声明式导航&编程式导航**

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

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

**4.1 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:

①this.$router.push('hash地址')

·跳转到指定hash地址，并增加一条历史记录

②this.$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

③this.$router.go(数值n)

·实现导航历史前进、后退

**4.2 $router.push**

调用this.$router.push( )方法，可以跳转到指定的hash地址，从而展示对应的组件页面。示例代码如下:



**4.3 $router.replace**

调用this.$router.replace( )方法，可以跳转到指定的 hash地址，从而展示对应的组件页面。

push和replace的区别:

· push 会增加一条历史记录

· replace不会增加历史记录，而是替换掉当前的历史记录

**4.4 $router.go**

调用this.$router.go( )方法，可以在浏览历史中前进和后退。示例代码如下:



**4.5 $router.go的简化用法**

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

①$router.back( )

·在历史记录中，后退到上一个页面

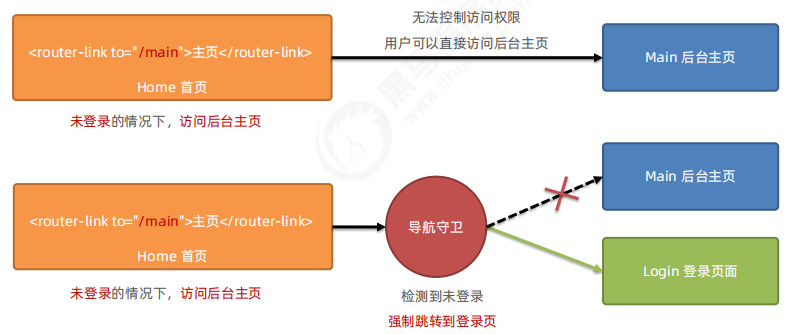
②$router.forward( )

·在历史记录中，前进到下一个页面

**在行内使用编程式导航跳转的时候, this必须要省略，否则会报错!**

**5.导航守卫**

导航守卫可以控制路由的访问权限。示意图如下:



**5.1 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:



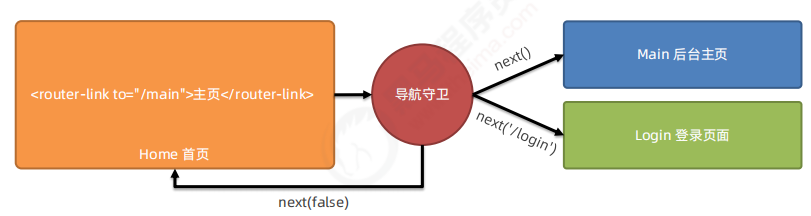
**5.2 守卫方法的3个形参**

全局前置守卫的回调函数中接收3个形参，格式为:



**5.3 next函数的3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:



当前用户拥有后台主页的访问权限，直接放行:next( )

当前用户没有后台主页的访问权限，强制其跳转到登录页面: next('/login')

当前用户没有后台主页的访问权限，不允许跳转到后台主页: next(false)

**5.4控制后台主页的访问权限**

