目录

[1 动态路径参数 1](#_Toc25059)

[2嵌套路由（子路由） 1](#_Toc5427)

[3路由跳转方式： 1](#_Toc21179)

[4命名视图 2](#_Toc28271)

1 动态路径参数

含义：将参数放到路径中，例子:{ path: '/user/:id', component: User }

注：动态路径参数以冒号开头，参数值会被设置到 this.$route.params，

通过$route.params.id进行访问，

注意，动态路径中不同参数访问的是同一个组件，第二次访问时此组件的生命周期钩子函数

不会调用了，所有一些方法不能直接放在钩子函数执行，需要做一些变形

如1）监听route对象 watch: {

$route(to, from) {

// 对路由变化作出响应...

}

}

2）使用 2.2 中引入的 beforeRouteUpdate 导航守卫：beforeRouteUpdate (to, from, next) { // react to route changes...

// don't forget to call next() }

高级匹配模式-1):表示参数（例子-/:id），2)后缀加问号表示可选参数（例子-/:id?）

3)后缀加星号\*，表示不定项参数0-n; 加+号表示多个参数1-n '/bar/baz'

（例子-/:id\*,加号一样，使用 /ces/ces,则路由里的参数 '/ces/ces' : '/ces/ces'）

4）自定义匹配-正则表达式[^\/]+，不符合的就无法放到参数里

更多见：https://github.com/pillarjs/path-to-regexp/tree/v1.7.0#parameters

2嵌套路由（子路由）

1）在一个组件A里面添加 <router-view></router-view>

2）配置A的路由，添加属性children:[放子组件路由]，在A页面访问子组件路径时会自动填充

到<router-view></router-view>

3路由跳转方式：

router-link(通过to属性)和router.push方式（本质上link也是调用push实现跳转）

1）Push方式this.$router.push(location, onComplete?, onAbort?)问号参数可选

location可以是字符串-即路径可以是路由对象，包括name,path,params属性,

**注意：**有path，params会被忽略，此时把参数写到路径(见下例，**也适用于to属性**)

要想写param，可以只写name,params或者{ path: 'register', query: { plan: 'private' },query会把参数以get形式的？id=1的传递参数

例子：带参数的跳转

this.$router.push({ path: `/RouterComponents/${val}`, })

或者

:to="{name:'RouterComponents',params:{id:110}}"

2)this.$router.replace(location, onComplete?, onAbort?)，不会添加历史记录-》

从C跳转到B,B执行replace到A,此时点击后退则会直接到C

1. )this.$router.go(n)，正整数前进，负整数后退

4命名视图

给<router-view name="a"></router-view>添加name属性，用于同级展示多个组件

1）首先定义一个组件A,A里面定义几个<router-view name=""></router-view>

name属性一定要有，除非只有一个，不写name属性默认为default

1. 定义A组件路由，此时不再使用component属性，使用components,一个对象，键值对就是router-view的name属性为键，对应的组件为值，

5重定向

在路由对象里面添加属性redirect，值可以是字符串路径，可以是路由对象，可以是方法，方法需要接收 目标路由 作为参数，return 重定向的 字符串路径/路径对象

例子： { path: '/a', redirect: { name: 'foo' }}

6别名

别名作用是让多个路径都匹配到同个路由，使用是路由对象添加alias属性，值为新的路径，可以是数组，可以是字符串

{path: '/a', component: A, alias:[ '/b']}

7路由跳转传参（不建议props过多使用）

基础见目录3，这里说一下路由对象的props属性，

7.1 布尔模式 props设置为 true，route.params 将会被设置为组件属性，注意每个组件都要设置一下props属性-针对嵌套路由，原本一个组件设置props:true就好，子组件就要按照名字来设置，如下：

7.1.1{ path: '/user/:id', component: User, props: true },

7.1.2{components: { default: User, sidebar: Sidebar },

props: { default: true, sidebar: false }}

7.2对象模式 ：如果 props 是一个对象，它会被按原样设置为组件属性。当 props 是静态的时候有用。

7.3函数模式 ；props可以是一个函数，返回一个对象参数，可以将参数变形