**05.动态路由匹配**

学习要点：

1.动态路由匹配

本节课我们来开始了解 Vue router 的动态路由匹配的功能。**一．动态路由匹配**

1. 所谓动态路由匹配：就是将不确定的参数进行路由映射到同一个组件上去；
2. 比如经典的：user?id=5，这个 5 就是动态的数值，最终路径：user/5；
3. 当然，复杂一些也是可以的，官网提供了两个案例，如下：

|  |  |  |
| --- | --- | --- |
| 模式 | 匹配路径 | $route.params |
| /user/:id | /user/5 | {id:5} |
| /user/:id/post/:name | /user/5/post/lee | {id:5, post:'lee'} |

1. 为了方便理解，创建 user 页面组件后，先采用 query 模式：user?id=5；

<template>

<h3>User Id : {{$route.query.id}}</h3>

</template>

1. 如果要使用动态路由，先需要在 path 里配置相关路由规则，然后再获取；

{

path: '/user/:id', name: 'User', component: User

},

<template>

<h3>User Id : {{$route.params.id}}</h3>

</template>

PS：至于$route 对象有哪些属性，而已在官网路由的 API 查询，相关地址如下：

PS：https://router.vuejs.org/zh/api/#路由对象属性

# 我们创建两个不同参数但同一个组件调用的链接，方便来回点击调试；

<router-link to="/user/5">user/5</router-link> |

<router-link to="/user/6">user/6</router-link>

1. 由于动态路由访问的都是同一组件不同参数，组件会被复用以提升效率；
2. 所以，此时的生命周期的钩子不会再被调用；那怎么监听路由的变化呢？
3. 可以通过 watch 监听对象，来监听$route(to, from)判断去处和来源；
4. 在 App.vue 中，创建 script，来监听动态路由的变化；

<script>

export default { watch : {

$route(to, from) { ***console***.log(to) ***console***.log(from)

}

}

}

</script>

1. 有时路由不存在会得不到任何结果，可以设置捕获所有路由或 404；
2. 将 path 设置为\*号，将会捕获任何没有得到匹配的路由；

{

path: '\*',

name: 'Home', component: Home

},

{

path: '/user-\*',

name: 'Home', component: Home

},