

## 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'}

4. 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>
```

5. 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：

<https://next.router.vuejs.org/zh/api/#routelocationnormalized>

6. 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>
```

7. 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；
8. 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？

```
// 初始化
created() {
  // 复用导致只被调用一次
  console.log('我被调用了? ')
}
```

9. 可以通过 `watch` 方法，来监听地址的判断，这里用官网手册的 `this.$watch()`;

```
// 初始化
created() {
  // 监听来去, this.$route 也可以写成 this.$route.params 缩小范围
  this.$watch(() => this.$route, (to, from) => {
    // 去哪里?
    console.log(to)
    // 从哪里来? previous
    console.log(from)
  })
}
```

PS: 直接用 `watch` 监听对象实现的方法如下:

```
watch : {
  $route(to, from) {
    console.log(to)
    console.log(from)
  }
}
```