

06. 嵌套路由

学习要点：

1. 嵌套路由

本节课我们来开始了解 Vue router 的嵌套路由功能。

一. 嵌套路由

1. 实际应用场景中，可能存在多种嵌套组件的应用界面，类似栏目分类；
2. 比如：新闻板块下有国内新闻、国外新闻、体育新闻、音乐新闻等；
3. 我们接着上节课 User 路由来分：用户个人档案和用户个人岗位两个子路由；
4. 下面是嵌套路由的匹配方式，第一种固定路由，二三两种动态路由；

模式	匹配路径
/news/music	/news/music
/user/:id/profile	/user/5/profile
/user/:id/posts	/user/5/posts

5. 我们只有做二三两种即可，先创建两个 User 组件的子路由组件；
6. 可以在 views 目录下创建 user 目录，并分别创建 UserProfile 和 UserPosts；

```
<template>
  <h3>UserProfile 个人档案组件 {{$route.params.id}}</h3>
</template>
```

```
<template>
  <h3>UserPosts 个人岗位： {{$route.params.id}}</h3>
</template>
```

7. 然后，我们在 route/index.js 配置路由规则，在 User 组件中配置子路由；

```
{
  path: '/user/:id',
  name: 'User',
  component: User,
  children : [
    {
      path : 'profile',
      component : UserProfile
    },
    {
      path : 'posts',
      component : UserPosts
    },
  ],
}
```

```
    ]  
  },
```

PS: 这里组件名称和路径设置的不一样，路由采用的是路径；

PS: 子路由不需要 / ，因为斜杠会被当成根路径；

8. 在 `User.vue` 这个父路由，需要使用视图渲染加载子路由；

```
<template>  
  <div>  
    <h3>User Id : {{$route.params.id}}</h3>  
    <router-view></router-view>  
  </div>  
</template>
```

PS: `<router-view>` 外层需要有一个元素标签，否则会报错；

9. 如果父路由，遇到没有可渲染的子路由，可以设置一个空路由；

```
{  
  path : '*',  
  component : List  
},
```