

关于vue路由有一个tip:

类似这种形式的 : 其实际访问的形式是 /user/xxx/children1 既是访问到对于的子页面, 然后修改不同的路径访问不同的children1

```
1  const router = [  
2    {  
3      path: '/user/:id',  
4      component: 'user',  
5      children:[  
6        {  
7          path: 'children1',  
8          component: 'children1'  
9        },  
10       {  
11         path: 'children2',  
12         component: 'children2'  
13       }  
14     ]  
15   }  
16 ]
```

一些应用程序的 UI 由多层嵌套的组件组成。在这种情况下, URL 的片段通常对应于特定的嵌套组件结构, 例如:

```
1  /user/johnny/profile          /user/johnny/posts  
2  +-----+                    +-----+  
3  | User          |            | User          |  
4  | +-----+ |            | | +-----+ |  
5  | | Profile    | | +-----> | | Posts      | |  
6  | |            | |            | |            | |  
7  | +-----+ |            | | +-----+ |  
8  +-----+                    +-----+  
9
```

通过 Vue Router, 你可以使用嵌套路由配置来表达这种关系。

接着上节创建的 app :

```

1 <div id="app">
2   <router-view></router-view>
3 </div>
4

```

```

1 const User = {
2   template: '<div>User {{ $route.params.id }}</div>',
3 }
4
5 // 这些都会传递给 `createRouter`
6 const routes = [{ path: '/user/:id', component: User }]
7

```

这里的 `<router-view>` 是一个顶层的 router-view。它渲染顶层路由匹配的组件。同样地，一个被渲染的组件也可以包含自己嵌套的 `<router-view>`。例如，如果我们在 User 组件的模板内添加一个 `<router-view>`：

```

1 const User = {
2   template: `
3     <div class="user">
4       <h2>User {{ $route.params.id }}</h2>
5       <router-view></router-view>
6     </div>
7   `,
8 }
9

```

要将组件渲染到这个嵌套的 router-view 中，我们需要在路由中配置 children：

```

1 const routes = [
2   {
3     path: '/user/:id',
4     component: User,
5     children: [
6       {

```

```

7      // 当 /user/:id/profile 匹配成功
8      // UserProfile 将被渲染到 User 的 <router-view> 内部
9      path: 'profile',
10     component: UserProfile,
11   },
12   {
13     // 当 /user/:id/posts 匹配成功
14     // UserPosts 将被渲染到 User 的 <router-view> 内部
15     path: 'posts',
16     component: UserPosts,
17   },
18 ],
19 },
20 ]
21

```

**注意，以 / 开头的嵌套路径将被视为根路径。这允许你利用组件嵌套，而不必使用嵌套的 URL。**

如你所见，children 配置只是另一个路由数组，就像 routes 本身一样。因此，你可以根据自己的需要，不断地嵌套视图。

此时，按照上面的配置，当你访问 /user/eduardo 时，在 User 的 router-view 里面什么都不会呈现，因为没有匹配到嵌套路由。也许你确实想在那里渲染一些东西。在这种情况下，你可以提供一个空的嵌套路径：

```

1  const routes = [
2    {
3      path: '/user/:id',
4      component: User,
5      children: [
6        // 当 /user/:id 匹配成功
7        // UserHome 将被渲染到 User 的 <router-view> 内部
8        { path: '', component: UserHome },
9
10       // ...其他子路由
11     ],
12   },
13 ]

```

