

## 第二十一课 路由元信息与组件缓存

### 学习目录

- 路由元信息
- 组件缓存

### 一. 路由元信息

有时候我们配置的路由需要定制化的配置参数,比如说不同的路由对应修改不同的页面文档标题, 或者不同路由对应的组件是否需要登录才能访问, 又或者有的组件需要缓存或者不缓存, 组件缓存我们等下再说, 因此路由元信息可以帮助我们完成这些工作。

```
const router = new VueRouter({  
  routes: [  
    {  
      path: '/book',  
      component: Book,  
      meta: {  
        title: '书籍'  
      }  
    }  
  ]  
})
```

一个路由匹配到的所有路由记录会存储在 route 路由信息对象中。因此，我们可以从路由记录中拿到 meta 路由元信息。

```
<template>

  <div>

    <h3>{{ $route.meta.title }}</h3>

  </div>

</template>
```

利用路由元信息以及我们之前了解的导航守卫，我们可以改变页面的标题。

```
router.beforeEach((to, from, next) => {

  window.document.title = to.meta.title;

  next();

});
```

## 二. 组件缓存

一般来说我们组件在使用路由切换的时候会刷新组件状态和数据，但是有时候用户操作组件的时候希望之前的操作在组件切换的时候可以被保留而不是被刷新。在 vue 中这种组件缓存依靠内置组件 keep-alive 来完成。

<keep-alive> 包裹组件时，会缓存组件实例，<keep-alive> 是一个抽象内置组件，它自身不会变成一个 DOM 元素，也不会成为被缓存组件的父级组件。

<keep-alive> 不能缓存函数式组件，因为函数式组件没有缓存的 vue 实例对象。

```
<keep-alive>
```

```
  <component :is="tabIndex"> </component>
```

```
</keep-alive>
```

```
<keep-alive>
```

```
  <router-view> </router-view>
```

```
</keep-alive>
```

被 keep-alive 包裹的动态组件或者 router-view 会缓存不活动的组件，假如在一次调用被缓存的组件，由于仍然使用之前缓存的组件数据，不用在次发送 ajax 请求获取数据，所以可以减少性能上的消耗。但是同样也存在一个问题就是被 keep-alive 包裹的组件我们请求的数据不会重新渲染页面，因此在使用 keep-alive 的时候可以根据实际情况来决定哪些组件需要缓存哪些不需要缓存。

利用 include、exclude 属性来指定需要缓存和不需要缓存的组件名称。

include 参数是字符串或正则表达式。只有名称匹配的组件会被缓存。

exclude 参数是字符串或正则表达式。任何名称匹配的组件都不会被缓存。

```
<keep-alive :include="['userWork','userHobbies']" :exclude="['userIncome']">
```

```
  <component :is="tabIndex"> </component>
```

```
</keep-alive>
```

可以利用 meta 路由元信息来决定路由配置中需要缓存的组件。

```
<router-view v-if="!$route.meta.keepAlive"></router-view>
```

```
<keep-alive>
```

```
<router-view v-if="$route.meta.keepAlive"></router-view>
```

```
</keep-alive>
```

组件缓存大多数情况下还是比较好用的，特别适合需要对用户操作进行保存、数据量较大长时间不需要刷新的图表组件等。

**谢谢观看！如果觉得课程还不错的话，记得给个好评！**

我是星星课堂老师：周小周