

第二十课 导航守卫

学习目录

- 导航守卫

一. 导航守卫

导航守卫顾名思义用来守卫导航是否显示 url 对应组件的作用，导航守卫可以用来通过跳转或取消的方式守卫导航，因此导航守卫也是比较常用的路由控制方法。导航守卫有全局导航守卫、单个路由导航守卫、单个组件导航守卫。

1.全局导航守卫

router.beforeEach 全局前置导航守卫

```
router.beforeEach((to, from, next) => {  
  
  console.log(to);  
  
  console.log(from);  
  
  next();  
  
})
```

router.beforeEach 守卫方法接收三个参数：

to: 即将要进入的目标路由对象

from: 当前导航正要离开的路由

next: 保证导航执行效果

next()方法不传递任何参数表示直接进入下一个管道中的导航。

`next(false)`取消当前导航。

`next('/book')`第一个参数如果传递 url 路径、path、name 等路径跳转参数，可以取消导航跳转到参数指定的导航。

`next` 方法能调用多次，但是必须是在业务逻辑判断中只能被执行一次。

```
router.beforeEach((to, from, next) => {  
  
  if (to.name !== 'Login' && !hasLogin) {  
  
    next({ name: 'Login' });  
  
  } else {  
  
    next();  
  
  }  
  
})
```

`router.afterEach` 全局后置导航守卫

全局后置导航守卫只能查看 `to` 和 `from`，不能修改 `next` 方法，因为这个时候导航已经完成了跳转。

```
router.afterEach((to, from) => {  
  
  console.log(to);  
  
  console.log(from);  
  
})
```

2. 单个路由导航守卫

我们还可以在路由配置里使用只对当前路由有作用的导航守卫。

```
const router = new VueRouter({  
  
  routes: [  
  
    {  
  
      path: '/book',  
  
      component: Book,  
  
      beforeEnter: (to, from, next) => {  
  
        console.log(to);  
  
        console.log(from);  
  
        next();  
  
      }  
  
    }  
  
  ]  
  
})
```

3.单个组件导航守卫

beforeRouteEnter

当该组件的对应路由被加载前调用，此时组件实例还没有构建，因此不能在这个钩子函数中直接使用 this，不过可以通过传一个回调给 next 来访问当前组件 this 实例对象。

```
beforeRouteEnter (to, from, next) {  
  
  next(vm => {  
  
    console.log(vm);  
  
  })  
  
}
```

```
}
```

beforeRouteUpdate

当前路由改变更新时或者组件被复用时调用，一般来说这里还可以在动态路径参数发生改变但是组件被复用的时候被调用，对于一个带有动态参数的路径 `/book/:id`，在 `/book/1` 和 `/book/2` 之间跳转的时候，由于会渲染同样的 `Book` 组件，因此组件实例会被复用。而这个组件导航守卫钩子函数就会在这个情况下被调用，这里可以访问 `this` 实例对象了。

```
beforeRouteUpdate (to, from, next) {
```

```
  this.title = to.params.title
```

```
  next();
```

```
}
```

beforeRouteLeave

导航离开该组件的对应路由时调用，一般用于表单组件或者其他询问用户是否需要退出组件中，这里可以访问 `this` 实例对象了。

```
beforeRouteLeave (to, from, next) {
```

```
  let tip = window.confirm('表单未提交，需要先保存表单数据么?')
```

```
  if (answer) {
```

```
    next();
```

```
  } else {
```

```
    next(false);
```

```
  }
```



学习前端，最快的进步是持续！

}

备注一下：一般来说导航守卫有解析顺序，如果把上述的导航守卫都用上的话，在我们导航跳转的时候会按照以下顺序进入响应的导航守卫钩子函数中。

- 1.在正要离开的组件里调用 `beforeRouteLeave` 导航守卫。
- 2.调用全局的 `beforeEach` 导航守卫。
- 3.在复用的组件里调用 `beforeRouteUpdate` 导航守卫。
- 4.在路由配置里调用 `beforeEnter`。
- 5.在进入的组件里调用 `beforeRouteEnter`。
- 6.调用全局的 `afterEach` 导航守卫。

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