



链滴

iview Admin-pro 鉴权篇 (1)

作者: [chidaihuashi](#)

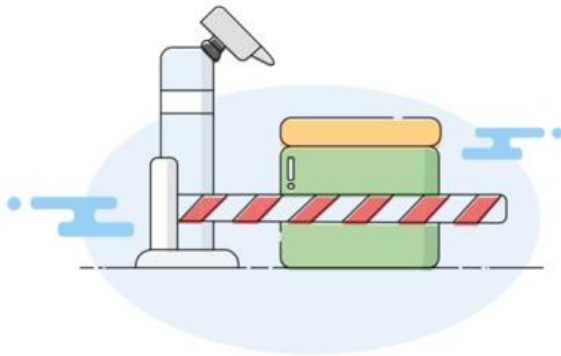
原文链接: <https://ld246.com/article/1602654806432>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

路由鉴权

路由鉴权是指，用户如果没有权限，无法在浏览器通过输入 URL 直接访问页面（一般会跳转到 403 面）：



403

抱歉，你无权访问该页面

[返回首页](#)

路由鉴权是严格意义上的安全，相比菜单栏鉴权，它是彻底无法访问页面，而不是通过隐藏入口。

约定

iView Admin Pro 对于鉴权有以下约定：

约定：用户登录后，其个人信息必须有 `access` 字段，且为数组。

解读：

1. 如果用户未登录，即用户还没有 `access` 信息，则但凡有设置鉴权的路由、菜单栏等都无法通过。
2. 用户登录后，它的 `access` 即为该用户拥有的权限，比如某用户为 `access: ['user']`，那就意味着有 `user` 的权限；比如是 `access: ['user', 'admin']`，那就代表拥有 `user` 和 `admin` 的权限。`access` 就会为系统鉴权的依据。

设置方法

设置路由鉴权，首先要每个页面的路由配置中，在 `meta` 里设置一个 `auth` 字段，然后在 `router.beforeEach` 中做判断。

示例一：用户未登录，则无法访问

以页面 `dashboard` 为例，先在路由的 `meta` 中配置权限：

```
// src/router/modules/dashboard.js
// 部分代码省略
{
  path: 'console',
  name: `${pre}console`,
```

```

    meta: {
      auth: true,
      title: '主控台'
    },
    component: () => import('@/pages/dashboard/console')
  }
}

```

这里因为只需要判断是否登录，所以将 `auth` 设置为了 Boolean，使用者也可以统一为数组。

然后在 `router.beforeEach` 中判断用户是否登录：

```

// src/router/index.js
// 部分代码省略

/**
 * 路由拦截
 * 权限验证
 */
router.beforeEach((to, from, next) => {
  // 判断是否需要登录才可以进入
  if (to.matched.some(_ => _.meta.auth)) {
    // 这里依据 token 判断是否登录，可视情况修改
    const token = util.cookies.get('token');

    if (token && token !== 'undefined') {
      next();
    } else {
      // 没有登录的时候跳转到登录界面
      // 携带上登陆成功之后需要跳转的页面完整路径
      next({
        name: 'login',
        query: {
          redirect: to.fullPath
        }
      });
    }
  } else {
    // 不需要身份校验 直接通过
    next();
  }
});

```

这里通过判断是否能从 Cookie 得到用户的 token，如果能得到，就准许访问，否则跳转到登录页。

示例二：用户是管理员，才能访问

同样以 `dashboard` 页面为例，先在路由的 meta 中配置权限：

```

// src/router/modules/dashboard.js
// 部分代码省略
{
  path: 'console',
  name: `${pre}console`,
  meta: {

```

```

    auth: ['admin'],
    title: '主控台'
  },
  component: () => import('@/pages/dashboard/console')
}

```

然后在 `router.beforeEach` 中判断用户是否登录且是否为管理员：

```

// src/router/index.js
// 部分代码省略

```

```

import { includeArray } from '@/libs/system';

```

```

/**
 * 路由拦截
 * 权限验证
 */

```

```

router.beforeEach(async (to, from, next) => {
  if (to.matched.some(_ => _.meta.auth)) {
    // 这里依据 token 判断是否登录，可视情况修改
    const token = util.cookies.get('token');

    if (token && token !== 'undefined') {
      // 获取用户权限，这里不能直接从 Vuex 的 user.info 读，因为数据还未准备好
      const userInfo = await store.dispatch('admin/db/get', {
        dbName: 'sys',
        path: 'user.info',
        defaultValue: {},
        user: true
      }, { root: true });
      const access = userInfo.access;
      const isPermission = includeArray(to.meta.auth, access);
      if (isPermission) {
        next();
      } else {
        next({
          name: '403'
        });
      }
    } else {
      // 没有登录的时候跳转到登录界面
      // 携带上登陆成功之后需要跳转的页面完整路径
      next({
        name: 'login',
        query: {
          redirect: to.fullPath
        }
      });
    }
  } else {
    // 不需要身份校验 直接通过
    next();
  }
}

```

});