[路由守卫、router路由钩子](https://www.cnblogs.com/qdwz/p/10969716.html)

在路由跳转的时候，我们需要一些权限判断或者其他操作。这个时候就需要使用路由的钩子函数。

定义：路由钩子主要是给使用者在路由发生变化时进行一些特殊的处理而定义的函数。

总体来讲vue里面提供了三大类钩子，两种函数  
1、全局钩子  
2、某个路由的钩子  
3、组件内钩子

两种函数：

Vue.beforeEach(*function*(*to*,*form*,*next*){}) */\*在跳转之前执行\*/*

Vue.afterEach(*function*(*to*,*form*))*/\*在跳转之后判断\*/*

**全局钩子函数**

顾名思义，它是对全局有效的一个函数

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

*let* token = router.app.$storage.fetch("token");

*let* needAuth = to.matched.some(*item* => item.meta.login); *//返回的就是当前路由匹配到的组件类*

    if(!token && needAuth) return next({path: "/login"});

    next();

});

**beforeEach函数有三个参数：**

* to router即将进入的路由对象
* from 当前导航即将离开的路由
* next Function,进行管道中的一个钩子，如果执行完了，则导航的状态就是 confirmed （确认的）；否则为false，终止导航。

afterEach函数不用传next()函数

**某个路由的钩子函数**

顾名思义，它是写在某个路由里头的函数，本质上跟组件内函数没有区别。

*const* router = new VueRouter({

    routes: [

      {

        path: '/login',

        component: Login,

        beforeEnter: (*to*, *from*, *next*) => {

*// ...*

        },

        beforeLeave: (*to*, *from*, *next*) => {

*// ...*

        }

      }

    ]

  })

**路由组件的钩子**

注意：**这里说的是路由组件！**

路由组件 属于 组件，但组件 不等同于 路由组件！所谓的路由组件：直接定义在router中component处的组件。如：

*var* routes = [

    {

    path:'/home',

    component:home,

    name:"home"

    }

]

在子组件中调用路由的钩子函数时无效的。

在官方文档上是这样定义的：

可以在组件内直接定义以下路由导航钩子

beforeRouteEnter

beforeRouteUpdate (2.2 新增)

beforeRouteLeave

这里简单说下钩子函数的用法：它是和data,methods平级的。

beforeRouteLeave(to, from, next) {

    next()

},

beforeRouteEnter(to, from, next) {

    next()

},

beforeRouteUpdate(to, from, next) {

    next()

},

data:{},

method: {}

在Vue2.0中的路由钩子主要是用来拦截导航，让它完成跳转或前取消，可以理解为路由守卫。

分为全局导航钩子，单个路由独享的钩子，组件内钩子。

三种 类型的钩子只是用的地方不一样，都接受一个函数作为参数，函数传入三个参数，分别为to,from,next。

其中next有三个方法

(1)   next(); *//默认路由*

(2)   next(false); *//阻止路由跳转*

(3)   next({path:'/'}); *//阻止默认路由，跳转到指定路径*