**导航钩子函数**

导航发生变化时，导航钩子主要用来拦截导航，让它完成跳转或取消

**1.执行导航钩子函数的位置：**

1.**router全局**

2.**单个路由**

3.**组件中**

**2. 导航钩子函数**

**router实例上**：beforeEach、afterEach方法

**单个路由中：**beforeEnter方法

**组件内的钩子：**beforeRouteEnte、beforeRouteUpdate、beforeRouteLeave方法

**3. 导航钩子函数接收的参数**

**to** 要进入的目标 路由对象-到哪去

**from** 正要离开导航的路由对象-从哪来

**next** 用来决定跳转或取消导航

**下面是全局钩子函数**beforeEach、afterEach**的例子：**

即router实例上的

在路由中

................

Vue.use(VueRouter)

let router = new VueRouter({.................})

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

console.log('beforeEach')

next()//调用此方法表示进入导航

// next(false)或者不调用表示取消进入导航

})

.............

上面代码当刷新页面的时候，执行beforeEach,beforeEach有一个回调函数如果要进入导航就要执行next()方法，如果不执行，就进入不了导航页面，在浏览器上的表现就是，home、document、about、user可以切换并打印出'beforeEach'字符串，但是不显示具体内容。

**1.全局钩子函数beforeEach可以用来判断用户是否登录**

当用户访问home、document、about、user任意一个导航页面的时候，

通过为每个导航路由添加meta:{login:true}的状态。

然后再在路由中：

................

routes: [

{

path:'/',

component:home,

meta:{index:0, **login:true** }

}

},

{

path: '/home',

component: home,

name: 'home',

alias:'/index'//为home页设置别名

},

{

path: '/about',

component: about,

children:[

{

path:'',//设置为空默认访问此组件

component:study,

meta:{index:2, **login:true** }

}

},

{

path:'work',

component:work,

name:'Work'

},

{

path:'hobby',

component:hobby

},

]

},

{

path: '/odocument',

components:{//一个路由对应多个组件

default:odocument,//默认组件

doc\_slider:slider,

doc\_footer:footer

},

meta:{index:1, **login:true** }

}

},

{

path: '\*',

//component: notFound

//重定向

//redirect:'/home'

//redirect:{path:'/home'}

//redirect:{name:'home'}

//动态重定向

redirect: (to) => {//to保存着一个对象-访问的路径的路由信息

//console.log(to);

switch(to.path){

case '/123':

return 'home';

break;

case '/456':

return '/odocument';

break;

default:

return '/about';

}

}

},

{

path: '/user/:tip?/:userId?',//动态路径的写法 问号表示匹配一次或者零次

component:user,

meta:{index:3，

**login:true** }

}

]

})

Vue.use(VueRouter)

let router = new VueRouter({.................})

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

if(to.meta.login){

next('/login')//登录页面

}else{

next()

}

})

.............

上面代码当访问home、document、about、user任意一个导航页面的时候，如果目标路由的meta的login对象为true.则表示还没有登录，

next('/login')表示重定向到/login

**2.全局钩子函数afterEach表示进入路由后的操作可以用来修改title**

在路由中：

..........................

routes: [

{

path:'/',

component:home,

meta:{

index:0,

**title:"home"**

}

},

{

path: '/home',

component: home,

name: 'home',

alias:'/index'//为home页设置别名

},

{

path: '/about',

component: about,

children:[

{

path:'',//设置为空默认访问此组件

component:study,

meta:{

index:2,

**title:'about'**

}

},

{

path:'work',

component:work,

name:'Work'

},

{

path:'hobby',

component:hobby

},

]

},

{

path: '/odocument',

components:{//一个路由对应多个组件

default:odocument,//默认组件

doc\_slider:slider,

doc\_footer:footer

},

meta:{

index:1,

**title:'document'**

}

},

{

path: '\*',

//component: notFound

//重定向

//redirect:'/home'

//redirect:{path:'/home'}

//redirect:{name:'home'}

//动态重定向

redirect: (to) => {//to保存着一个对象-访问的路径的路由信息

//console.log(to);

switch(to.path){

case '/123':

return 'home';

break;

case '/456':

return '/odocument';

break;

default:

return '/about';

}

}

},

{

path: '/user/:tip?/:userId?',//动态路径的写法 问号表示匹配一次或者零次

component:user,

meta:{

index:3,

**title:'user'**

}

}

]

})

//进入导航之前执行的导航钩子

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

console.log('beforeEach')

next()

})

//进入导航之后使用的钩子函数

**router.afterEach((to,from)=>{**

if(to.meta.title){

window.document.title=to.meta.title

}

}**)**

上面代码进入每个导航之后如果有title就会将window.document.title设置成

定义的title

**下面是单个路由钩子函数**beforeEnter**的例子：**

beforeEnter表示在具体某一个路由执行的钩子函数，而不是向上面一样全局的。

例：进入user之前的beforeEnter

在路由中：

.......................

{

path: '/user/:tip?/:userId?',//动态路径的写法 问号表示匹配一次或者零次

component:user,

meta:{

index:3,

title:'user'

},

**beforeEnter(**to,from,next){

console.log('beforeEnter')

next();//调用此方法表示进入导航

}

}

.....................................

上面的代码在切换到user之外的导航的时候不会打印'beforeEnter'，

在切换到user时，会打印'beforeEnter'，但是只有在调用了next()方法后才会进入user页面