在login.vue组件中：

通过给元素添加ref 属性，就可以在vue实例中通过

this.**$refs**.name 找到这个元素

在login.vue中:

<template>

<div class='login\_box'>

<h2>登陆</h2>

<form @submit.prevent='sendLogin' >

<div><input placeholder="请输入用户名" type="text" name="user" **ref**='**userInput**'/></div>

<div><input placeholder="请输入密码" type="password" name="password" /></div>

<div ><input class="btn\_login" type="submit" value="一键登入" /></div>

</form>

<p>首页>>></p>

</div>

</template>

<script>

export default {

methods:{

sendLogin(){

//完成登录由于没有后端这里不用ajax发送请求用localstorage

console.log(this.**$refs**.**userInput**)

}

}

}

</script>

C:\Users\Administrator\Desktop\aad.png

接下来将取到的用户名和密码都放到localstorage里

在login.vue里：

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

<template>

<div class='login\_box'>

<h2>登陆</h2>

<form @submit.prevent='sendLogin' >

<div><input placeholder="请输入用户名" type="text" name="user" ref='userInput'/></div>

<div><input placeholder="请输入密码" type="password" name="password" ref='passInput'/></div>

<div ><input class="btn\_login" type="submit" value="一键登入" /></div>

</form>

<p>首页>>></p>

</div>

</template>

<script>

export default {

methods:{

sendLogin(){

//完成登录由于没有后端这里不用ajax发送请求用localstorage

//console.log(this.$refs.userInput.value)

//console.log(this.$refs.passInput.value)

//将用户名和密码存到localstorage里

let **username**=this.$refs.userInput.value

let **password**=this.$refs.passInput.value

**this.$local.save**('name',{

login:'true',//表示已登录状态

username:username,

password:password

})

//完了跳转到首页

this.$router.push('/')

}

}

}

</script>

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

上面的代码在点击提交后，用户名和密码就会保存到localstorage，

接着在用编程式导航的push方法跳转到指定的首页

跳转到首页后，此时时已登录状态，所以登录按钮消失，头像显示。

在home.vue中：

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

<template>

<div>

<div class='home-title-box'>

<p class="img\_logo\_box" >

<img src='../assets/icon\_todo\_tasks.png'/>

<span>By Tomato</span>

</p>

<router-link to='/login' **v-if='*!isLogin*'** tag='button'>登录</router-link>

<img class='person\_logo' **v-if='*isLogin*'** src='../assets/icon\_after\_service.png'/>

</div>

<h1 class="home\_title">vue-router学习</h1>

<ul class='home\_list clearft'>

<li>我的项目</li>

<li style='float:right;'>我的文档</li>

</ul>

</div>

</template>

<script>

export default {

data () {

return {

***isLogin***:false

}

},

created(){//

**let info=this.$local.fetch('name')**

this.***isLogin***=info.login

}}

</script>

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

[**created**](https://vuefe.cn/v2/api/#created)**钩子函数表示**实例已经创建完成之后被调用。在这一步，实例已完成以下的配置：数据观测(data observer)，属性和方法的运算， watch/event 事件回调。

**我的项目**和我的和**工作台**必须是登录状态才能显示

**通过导航钩子函数来实现**在路由中：

{//后台管理页面（组件）的入口

path: '/management',

component: layout,

name: 'management',

children:[

{

path:'/project',

component:project,

name:'project',

**meta:{**

**login:true/\*用导航钩子函数每次进入判断是否登录过\*/**

**}**

},

{

path:'/workbench',

component:workbench,

name:'workbench',

**meta:{**

**login:true/\*用导航钩子函数每次进入判断是否登录过\*/**

**}**

},

{

path:'/doc',

component:doc,

name:'doc',

**meta:{**

**login:false/\*用导航钩子函数每次进入判断是否登录过\*/**

**}**

}

]

},

/\*设置全局的导航钩子函数\*/

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

let bl=to.matched.some((item) => item.meta.login) //console.log(bl)

if(bl){

let info=**router.app**.$local.fetch('name')

if(info.login){

next()

}else{

next('/login')

}

}else{

next()

}

})

export default router

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

为management下面的每个子组件添加一个登录状态，如果需要登录**login:true**为，否则

**login：false。**

**注意：由于**project、workbench、doc、都是子路由，为了保险在判断登录的时候，需要判断父路由和子路由是否都有登录状态。

实现保险在判断的方法：

如上面代码：

to目标对象里有一个matched数组对象，这里面存放的是自己的路由信息对象，和父路由信息对象

例如访问'management下面的project，那么project的to的matched数组对象里就存放着

'management和project（它自己的）路由信息

上面代码to.matched拿到了目标对象和它的父对象，再将to.matched用some方法遍历出

来。如果有一项中的meta.login的状态为true,就表示需要登录，再判断需要登录的是否，

已经登录过了（即用户名密码已经保存在localStorage中了），用fetch方法取vue实例中name对象的login状态值。**router.app.$local.fetch('name').login.**注意 ：由于在箭头函数里没有this,所以这里使用**router.app，router.app**就指向根实例，所以这样用。

在没有登录之前，访问的地址被记录，登陆后直接跳到

没登录之前访问的地址。用query查询字符串。

在路由中：

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

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

let bl=to.matched.some((item) => item.meta.login);

console.log(to);

if(bl){

let info=router.app.$local.fetch('name');

if(info.login){

next();

}else{

router.push({

path:'/login',

query:{

**redirect:to.path**.slice(1)//将之前的路径保存在查询字符串里

}

})

}

}else{

next();

}

})

export default router

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

上面的代码当用户进入登录页面，将之前访问的目标地址**to.path**保存在查询字符串里，不要/所以截取一下

例如下面直接访问project在没有登录时会跳到login页面，通过上面的代码，project会被记录在根实例$toute.query对象上

http://localhost:8080/login?redirect=project

在login代码中：

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

export default {

methods:{

sendLogin(){

//完成登录由于没有后端这里不用ajax发送请求用localstorage

//console.log(this.$refs.userInput.value)

//console.log(this.$refs.passInput.value)

//将用户名和密码存到localstorage里

let username=this.$refs.userInput.value

let password=this.$refs.passInput.value

console.log(this.$route)

this.$local.save('name',{

login:'true',//表示已登录状态

username:username,

password:password

})

//完了跳转到登陆之前的页面

**let redirect=this.$route.query.redirect**

this.$router.push({

path:'/'+redirect

})

},

goHome(){

this.$router.push('/')

}

}

}

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

上面的代码当登录完成后，读取之前访问的记录在**this.$route.query.redirect**中

然后用编程式导航的push方法跳到对应页面。

最后优化：在没登录之前没有访问必须登录才能显示的组件，在登录后就让他默认跳到后台project页面。

login.vue中：

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

//完了跳转到登陆之前的页面

let redirect=this.$route.query.redirect

**if(**!redirect**){**

this.$router.push({

path:**'/project'**

})

**}else{**

this.$router.push({

path:**'/'+redirect**

})

**}**

},

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

上面代码当拿到的redirect的记录为空的时候，登录成功后就直接跳到**project**组件中。