路由：vue-router

1.

<script src="https://unpkg.com/vue/dist/vue.js"></script>

<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>

<div id="app">

<p>

<!-- 使用 router-link 组件来导航. -->

<!-- 通过传入 `to` 属性指定链接. -->

<!-- <router-link> 默认会被渲染成一个 `<a>` 标签 -->

<router-link to="/one">one</router-link>

<router-link to="/two">two</router-link>

</p>

<!-- 路由出口 -->

<!-- 路由匹配到的组件将渲染在这里 -->

<router-view></router-view>

</div>

<script>

var one = {

template: '<div>One视图</div>',

}

var two = {

template: '<div>Two视图</div>'

}

var routes = [

{

path: '/one',

component: one,

},

{

path: '/two',

component: two,

}

];

var router = new VueRouter({

routes: routes

});

var vm = new Vue({

router: router,

el: '#app',

});

</script>

**动态路由**

核心代码

methods: {

toNameClick: function () {

this.$router.push({

name: 'user',

params: {

id: '456'

}

});

},

toPathClick: function () {

this.$router.push({

path: '/user/789',

query: {

id: '101112'

}

});

}

**嵌套路由**

<div id="app" >

<p>我是#app中的内容</p>

<!-- 路由出口 -->

<!-- 路由匹配到的组件将渲染在这里 -->

<!-- 这里为sup路由的渲染出口 -->

<router-view></router-view>

</div>

<script type="text/x-template" id="sup">

<div>

<p>我是#sub中的内容</p>

<router-link to="/sub1">sub1</router-link>

<router-link to="/sub2">sub2</router-link>

<!-- 路由出口 -->

<!-- 路由匹配到的组件将渲染在这里 -->

<!-- 这里为sup子路由的渲染出口,即sub1、sub2 -->

<router-view></router-view>

</div>

</script>

<script>

var sup = {

template: '#sup'

}

var sub1 = {

template: "<div>我是sub1</div>"

}

var sub2 = {

template: "<div>我是sub2</div>"

}

var routes = [

{

path: '/',

component: sup,

children: [

{

path: 'sub1',

component: sub1

},

{

path: 'sub2',

component: sub2

}

]

}

];

var router = new VueRouter({

routes: routes

});

var vm = new Vue({

router: router,

el: '#app',

});

</script>

**高级用法**

1. 路由组件实例复用，使用beforeRouterUpdate(to,from,next){} 导航守卫方式来监听组件的复用。
2. 同级多路由展示 可以为<router-view></router-view> 添加name属性
3. 添加导航守卫监听页面变化。