

# 路由基础

- 什么是前端路由？

答案：路由是根据不同的url地址展示不同的内容或页面。前端路由就是把不同路由对应的不同的内容或页面的任务交给前端来做，之前是通过服务端根据url的不同返回不同的页面实现的。

- 前端路由有什么优点和缺点？

答案：

1. 优点：用户体验好，不需要每次都从服务器全部获取，快速展现给用户；
2. 缺点：不利于SEO，使用浏览器的前进，后退键的时候会重新发送请求，没有合理的利用缓存；单页面无法记住之前滚动的位置，无法在前进，后退的时候记住滚动的位置

=====

## vue-router

- vue-router 用来构建SPA
- 或者 `this.$router.push({path: ""})`
- 

=====

## 路由几种模式

- 动态路由匹配
  1. 什么是动态路由？

模式	匹配路径	\$route.params
<code>/user/:username</code>	<code>/user/evan</code>	<code>{ username: 'evan' }</code>
<code>/user/:username/post/:post_id</code>	<code>/user/evan/post/123</code>	<code>{ username: 'evan', post_id: 123 }</code>

```
//router.js中
1. 第一种 mode: 'hash'
export default new Router({
  mode: 'hash',      //没写也是默认这个模式
```

```

    routes: [
      {
        path: '/goods/:goodsId/user/:name',
        name: 'GoodsList',
        component: GoodsList
      }
    ]
  })

```

在浏览器里:

localhost:8080/#/goods/abc1818/user/admin

2. 第二种 mode: 'history'

```

export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods/:goodsId/user/:name',
      name: 'GoodsList',
      component: GoodsList
    }
  ]
})

```

在浏览器里:

localhost:8080/goods/abc1818/user/admin

- 嵌套路由: 可以实现同一个页面里不同菜单的切换

路由嵌套

```

export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods',
      name: 'GoodsList',
      component: GoodsList,
      children: [
        {
          path: 'title',
          name: 'title',
          component: Title
        },
        {
          path: 'img',
          name: 'img',

```

```

        component: Image
      }
    ]
  }
]
})

```

如果某个路由中二级路由，那么需要在这个路由页面里使用<router-view>来渲染  
这里在localhost:8080/goods 页面里需要有<router-view>

在GoodsList.vue中

```

<template>
  <div>
    这是商品列表页面
    <router-link to="/goods/title">显示商品标题</router-link>
    <router-link to="/goods/img">显示商品图片</router-link>
    <div>
      <router-view></router-view>
    </div>
  </div>
</template>

```

- 编程式路由

1. 什么是编程式路由？

答案：通过 js 来实现页面的跳转

```
$router.push('name')
```

```
$router.push({path: "name"})
```

```
$router.push({path: "name?a=123"}) 或者 $router.push({path: "name", query:
{a: 123}})
```

```
$router.go(1)
```

```

export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods',
      name: 'GoodsList',
      component: GoodsList,
      children: [
        {
          path: 'title',
          name: 'title',

```

```

        component: Title
      },
      {
        path: 'img',
        name: 'img',
        component: Image
      }
    ]
  },
  {
    path: '/cart',
    component: Cart
  }
]
})

```

在GoodsList.vue中

```

<template>
  <div>
    这是商品列表页面
    <router-link to="/goods/title">显示商品标题</router-link>
    <router-link to="/goods/img">显示商品图片</router-link>
    <div>
      <router-view></router-view>
    </div>
    <router-link to="/cart">跳转到购物车页面</router-link>
    <button @click="jump">button - 跳转到购物车页面</button>
  </div>
</template>

<script>
  export default {
    methods: {
      jump() {
        //第一种方式
        this.$router.push('/cart');
        //第二种方式
        this.$router.push({path: '/cart'});
        //那么如何传参呢?
        this.$router.push({path: '/cart?goodsId=123'});
        //点击Button, 浏览器倒退两步
        this.$router.go(-2);
      }
    }
  }
</script>

```

传递的参数，在页面中如何获取到呢？

在Cart.vue中：

```
<template>
  <div>
    这是购物车页面
    <span>{{ $route.query.goodsId }}</span>      // 123
  </div>
</template>
```

- 命名路由和命名视图

1. 给路由定义不同的名字，根据名字进行匹配
2. 给不同的router-view定义名字，通过名字进行对应组件的渲染

```
export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods',
      name: 'GoodsList',
      component: GoodsList,
    },
    {
      path: '/cart',
      name: 'cart',
      component: Cart
    }
  ]
})
```

在GoodsList.vue中

```
<template>
  <div>
    这是商品列表页面
    <router-link to="/goods/title">显示商品标题</router-link>
    <router-link to="/goods/img">显示商品图片</router-link>
  </div>
  <router-view></router-view>
</div>
  <router-link :to="{name: 'cart'}">跳转到购物车页面</router-link>
</div>
</template>
```

如果是动态的路由，如果通过命名路由实现

```
export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods',
      name: 'GoodsList',
      component: GoodsList,
    },
    {
      path: '/cart/:cartId',
      name: 'cart',
      component: Cart
    }
  ]
})

<template>
  <div>
    这是商品列表页面
    <router-link to="/goods/title">显示商品标题</router-link>
    <router-link to="/goods/img">显示商品图片</router-link>
  </div>
  <router-view></router-view>
</div>
  <router-link :to="{name: 'cart', params: {cartId: 123}}">跳转到购物车页
  面</router-link>
</div>
</template>
```

给不同的router-view定义名字，通过名字进行对应组件的渲染

App.vue中，相当于把页面分成3部分，比如第一个是导航，第二个是文章标题，第三个是文章图片，在路由中需要分别加载3个组件

```
<template>
  <div id="app">
    <router-view></router-view>
    <router-view name="title"></router-view>
    <router-view name="img"></router-view>
  </div>
</template>
```

```
router.js中
export default new Router({
  mode: 'history',
  routes: [
    {
      path: '/goods',
      name: 'GoodsList',
      components: { // component => components
        default: GoodsList,
        title: Title,
        img: Image
      }
    },
    {
      path: '/cart/:cartId',
      name: 'cart',
      component: Cart
    }
  ]
})
```