

1. 理解react-router

react的一个插件库
专门用来实现一个SPA应用
基于react的项目基本都会用到此库

2. 几个重要问题

1). SPA应用

单页web应用（single page web application, SPA）
整个应用只有一个完整的页面
点击页面中的链接不会刷新页面，本身也不会向服务器发请求
当点击路由链接时，只会做页面的局部更新
数据都需要通过ajax请求获取，并在前端异步展现（同时进行的）

2). 路由

1. 什么是路由？
一个路由就是一个映射关系（key:value）
key:为路由的路径 value:function/component
2. 路由分类
后台路由：node服务端的路由，key也是路由路径，value是一个函数function，用来处理客户端提交的请求并返回一个响应数据
前台路由：浏览器端的路由，value是component，当请求的路由是path时，浏览器就会更新显示对应的组件
3. 后台路由
 - * 注册路由：router.get(path, function(req, res))
 - * 当node接收到一个请求时，根据请求路径找到匹配的路由，调用路由中的函数来处理请求，返回响应数据
- * 前端路由
 - * 注册路由：<Route path="/about" component={About}>
 - * 当浏览器的hash变为#about时，当前路由组件就会变为About组件

3. react-router的基本使用

1). 下载

```
npm install react-router --save  
npm install react-router-dom --save 功能比较全
```

2). 路由模式

hash HashRouter （hash 模式 带#号 刷新时候页面不会丢失）
browser BrowserRouter （历史记录模式 没有带#号 他是通过记录历史记录api来进行路由切换的 刷新会丢失 本地模式不会）

3). 使用路由模式

在index.js引用路由模块
路由模式包裹包裹根组件

4). 配置 Route

引用 Route
<Route path="/about" component={About}>

5). 路由导航 NavLink、Link

引用NavLink、Link
<NavLink to="/about"></NavLink>

6). 路由规则 Switch

引用Switch
<Switch></Switch>

7). 路由重定向 Redirect

引用Redirect
<Redirect to="/about" />

8). 路由传参

1.params
<Route path="/home/news/detail/:id" component={Detail} />
方式一：
<NavLink to="/home/news/detail/\${id}"></NavLink>
方式二：
this.props.history.push(`/home/news/detail/\${id}`)
接收：this.props.match.params
优势：刷新地址，参数依然存在
缺点：只能传递字符串，不适合过多参数传递

2.query
<NavLink to={pathname:"/home/news/detail",query:{id:id}}></NavLink>
接收：this.props.location.query
缺点：刷新地址，参数不存在

9). 路由前进、后退

```
//前进  
this.props.history.goForward();  
//后退  
this.props.history.goBack();
```

4. 优化Link组件

1). NavLink.js

```
import React from 'react'  
import {Link} from 'react-router'  
export default function NavLink(props) {  
  return <Link {...props} activeClassName="active"/>  
}
```