

路由 2.x 对应 react, react-dom 15.x

1. 安装路由

```
# cnpm i react-router -S
```

2. 查看路由版本信息

```
npm info react-router
```

3. 安装指定版本router, react, react-dom

```
cnpm i react-router@2.8.1 react@15.6.2 react-dom@15.6.2 -S
```

下一步（在用户代码片段设置常用的react代码片段）

4. 在src下新建components文件夹，将主组件APP移入文件夹，文件夹下再新建About, Repos文件，分别设置代码片段

5. 在index.js内引入三个组件，再从react-router引入路由，渲染路由

6. 运行npm run dev

掌握九个内容：

1. 基本路由渲染

index.js文件，Router：整个路由，Route当前路由

```
import { Router, Route, hashHistory } from 'react-router' 50.8K (gzipped: 13.4K)

class Index extends React.Component {
  render() {
    return (
      <Router history={hashHistory}>
        <Route path="/" component={App}/>
        <Route path="/about" component={About}/>
        <Route path="/repos" component={Repos}/>
      </Router>
    )
  }
}

ReactDOM.render(<Index/>, document.getElementById('app'));
```

2. 添加一个链接

App.js文件，从react-router引入Link, 设置

```
index.js App.js x
src > components > App.js > App > render
1 import React, { Component } from 'react'; 8K (gzipped: 3.3K)
2 import { Link } from 'react-router'
3 export default class App extends Component {
4   render() {
5     return (
6       <div>
7         <h1>React router V2.x</h1>
8         <ul>
9           <li><Link to="/about">About</Link></li>
10          <li><Link to="/repos">Repos</Link></li>
11        </ul>
12      </div>
13    )
14  }
15 }
```

3. 基本嵌套路由

index.js文件，标签嵌套

```
index.js x App.js
src > index.js > index > render
10
11 import { Router,Route,hashHistory } from 'react-router' 50.8K (gzipped: 13.
12
13
14 class Index extends React.Component{
15   render(){
16     return [
17       <Router history={hashHistory}>
18         <Route path="/" component={App}>
19         <Route path="/about" component={About}/>
20         <Route path="/repos" component={Repos}/>
21       </Router>
22     ]
23   }
24 }
25
26
27 ReactDOM.render(<Index/>,document.getElementById('app'));
```

App.js 文件，子组件占位渲染

```
index.js x App.js x
src > components > App.js > App > render
1 import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2 import { Link } from 'react-router'
3 export default class App extends Component{
4   render(){
5     console.log(this.props);
6     return (
7       <div>
8         <h1>React router V2.x</h1>
9         <ul>
10           <li><Link to="/about">About</Link></li>
11           <li><Link to="/repos">Repos</Link></li>
12         </ul>
13         <h3>子组件占位渲染</h3>
14         {this.props.children}
15       </div>
16     )
17   }
18 }
```

4. 激活状态链接

css文件设置

```
.active{
  color: red
}
```

components文件夹下新建NavLink, js, 引入Link, 设置

...this.props 展开式参数传递, 就是在调用时把父组件的所有属性传递给当前的子组件

```
index.js x App.js x NavLink.js x
src > components > NavLink.js > NavLink > render
1 import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2 import { Link } from 'react-router'
3 export default class NavLink extends Component{
4   render(){
5     return (
6       <Link {...this.props} activeClassName="active"/>
7     )
8   }
9 }
```

App.js文件原先引入的Link就不需要了, 直接引入NavLink, 将标签Link换成NavLink

```
index.js  NavLink.js  App.js  x
src > components > App.js > App > render
1  import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2  import NavLink from './NavLink'
3
4
5  export default class App extends Component{
6    render(){
7      console.log(this.props);
8      return (
9        <div>
10         <h1>React router V2.x</h1>
11         <ul>
12           <li><NavLink to="/about">About</NavLink></li>
13           <li><NavLink to="/repos">Repos</NavLink></li>
14         </ul>
15         <h3>子组件占位渲染</h3>
16         {this.props.children}
17       </div>
18     )
19   }
20 }
```

5. 参数传递操作

设参在路由，在components下新建Repo.js文件，再在index.js引入组件

```
index.js  NavLink.js  App.js
src > index.js > Index > render
10
11 import { Router,Route,hashHistory } from 'react-router' 50.8K (gzipped: 13.4K)
12
13
14 class Index extends React.Component{
15   render(){
16     return (
17       <Router history=(hashHistory)>
18         <Route path="/" component={App}>
19         <Route path="/about" component={About}/>
20         <Route path="/repos" component={Repos}/>
21         <Route path="/repos/:userName/:repoName" component={Repo}/>
22       </Route>
23     </Router>
24   )
25 }
26
27 ReactDOM.render(<Index/>,document.getElementById('app'));
```

在Repos.js地址上传参

```
index.js  App.js  Repos.js  x
src > components > Repos.js > Repos > render
1  import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2  import NavLink from './NavLink'
3  export default class Repos extends Component{
4    render(){
5      return (
6        <div>
7          Repos
8          <hr/>
9          <ul>
10           <li><NavLink to="/repos/reactjs/react-router">React Router</NavLink></li>
11           <li><NavLink to="/repos/vuejs/vue-router">Vue Routers</NavLink></li>
12         </ul>
13       </div>
14     )
15   }
16 }
```

在Repo.js组件接参，解构赋值

```
index.js  App.js  Repos.js  x
src > components > Repos.js > Repo > render
1  import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2  export default class Repo extends Component{
3    render(){
4      const {userName,repoName} = this.props.params;
5      return (
6        <div>
7          Repo
8          <hr/>
9          <div>
10           <div>userName: {userName}</div>
11           <div>repoName: {repoName}</div>
12         </div>
13       )
14     }
15   }
```

6. 增强嵌套路由

index.js文件，用Repo标签嵌套repo

```
index.js  App.js  Repos.js  Repo.js
src > index.js > Index > render
8   import Repo from './components/Repo'
9
10  import style from './main.scss';
11
12  import { Router,Route,hashHistory } from 'react-router' 50.8K (gzipped: 13.4K)
13
14
15  class Index extends React.Component{
16  ... render(){
17  ...   return (
18  ...     <Router history={hashHistory}>
19  ...       <Route path="/" component={App}>
20  ...         <Route path="/about" component={About}/>
21  ...         <Route path="/repos" component={Repos}>
22  ...           <Route path="/repos/:userName/:repoName" component={Repo}/>
23  ...         </Route>
24  ...       </Route>
25  ...     </Router>
26  ...   )
27  ... }
28
29  ReactDOM.render(<Index/>,document.getElementById('app'));
```

在Repos.js文件，this.props.children占位显示

```
index.js  App.js  Repos.js  Repo.js
src > components > Repos.js > Repos > render
1   import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2   import NavLink from './NavLink'
3   export default class Repos extends Component{
4   ... render(){
5   ...   return (
6   ...     <div>
7   ...       Repos
8   ...       <hr/>
9   ...       <ul>
10  ...         <li><NavLink to="/repos/reactjs/react-router">React Router</NavLink></li>
11  ...         <li><NavLink to="/repos/vuejs/vue-router">Vue Router</NavLink></li>
12  ...       </ul>
13  ...       {this.props.children}
14  ...     </div>
15  ...   )
16  ... }
17
18  }
```

7. 默认页面路由 8. 默认页面链接

App.js文件设置

```
<ul>
  <li><NavLink to="/">Home</NavLink></li>
```

components文件夹下新建Home.js, 回到index.js引入Home, 在react-router引入IndexRoute, 设置Home，注意不要写path路径

```
index.js  App.js  Repos.js  Repo.js
src > index.js > Index > render
5   import App from './components/App'
6   import About from './components/About'
7   import Repos from './components/Repos'
8   import Repo from './components/Repo'
9   import Home from './components/Home'
10
11  import style from './main.scss';
12
13  import { Router,Route,hashHistory,IndexRoute } from 'react-router'
14
15
16  class Index extends React.Component{
17  ... render(){
18  ...   return (
19  ...     <Router history={hashHistory}>
20  ...       <Route path="/" component={App}>
21  ...         <IndexRoute component={Home}/>
22  ...         <Route path="/about" component={About}/>
23  ...         <Route path="/repos" component={Repos}>
24  ...           <Route path="/repos/:userName/:repoName" component={Repo}/>
25  ...         </Route>
26  ...       </Route>
27  ...     </Router>
```

但是会发现Home一直高亮显示

回到App.js，设置onlyActiveOnIndex

```

<ul>
  <li><NavLink to="/" onlyActiveOnIndex>Home</NavLink></li>

```

9. 导航链接操作

Repos.js文件，设置一个form表单，在1号位设置函数，注意结束不要写逗号分割

```

Repos.js
src > components > Repos.js > Repos > handlerSubmit
1  import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2  import NavLink from './NavLink'
3  export default class Repos extends Component{
4
5    handlerSubmit = ()=>{
6      |
7    }
8
9    render(){
10     return (
11       <div>
12         Repos
13         <hr></hr>
14         <ul>
15           <li><NavLink to="/repos/reactjs/react-router">React Router</NavLink></li>
16           <li><NavLink to="/repos/vuejs/vue-router">Vue Router</NavLink></li>
17         </ul>
18       </div>
19       <form onSubmit={this.handlerSubmit}>
20         <input type="text" ref="userName" placeholder="请输入用户名称" />
21         <input type="text" ref="repoName" placeholder="请输入资源名称" />
22         <button type="submit">提交</button>
23       </form>

```

引入hashHistory

```

Repos.js  index.js  Repos.js
src > components > Repos.js > Repos > handlerSubmit
1  import React,{Component} from 'react'; 8K (gzipped: 3.3K)
2  import NavLink from './NavLink'
3  import {hashHistory} from 'react-router'
4  export default class Repos extends Component{
5
6    handlerSubmit = (event)=>{
7      const uName = this.refs.userName.value;
8      const rName = this.refs.repoName.value;
9      const gotoUrl = `/repos/${uName}/${rName}`;
10     this.props.history.push(gotoUrl)
11     event.preventDefault();
12
13   }
14

```