

3-2 4.0基本路由功能DEMO实现-配置化

2018年11月23日

15:46

- Home1
- About1
- Topics1

this is About page.

【将Route路由提取到一个单独的JS文件中】

配置化实现路由功能

1. 创建Home.js内部写上ul->li->Link导航组件，并在想要显示对应路由内容的区域写上(this.props.children)即会加载调用Home.js组建时内部传递的信息
2. 创建router.js，最外层用HashRouter包裹，第二级用Home组件包裹，内部写对应路由Route(path路由路径匹配、及component属性渲染的组件)
3. 执行流程，index.js渲染Router组件时，由于最外层是HashRouter，内部是Home组件故加载Home组件内容，并通过(this.props.children)得到在调用Home组件时内部的信息。

嵌套路由

1. 如想实现在Main组件中的嵌套路由，需要在Main组件中添加(this.props.children)从而渲染对应的内部信息，还需要添加Link组件以进行跳转
2. 之后在router.js中对应调用该组件的Route组件中，删除component属性，添加render属性进行页面渲染，render属性内应是一个函数，返回Main组件（内部带有Route属性以进行路由渲染）
3. 注意点：
 - a. 调用Main组件的Route不能添加exact属性，因为如果添加exact属性，当点击内部组件路由到对应url时由于与外部Route的path不是完全匹配，故将不会显示
 - b. 调用Main组件的Route内部render函数如果添加()=>{}则需要再函数体内写return，因为{}表示函数体，内部的函数将被执行，返回组件需要写return；如果不添加大括号则直接写返回的组件即可，ES6箭头函数默认箭头后面的内容是return的
 - c. 调用Main组件的Route内部Main里的Route组件的path需要包含外部Route的path作为一级路由，如外部是/main内部需要是/main/xxx，如果不以外部Route作为一级路由则不会走外部的Route到内部Route内容

实例代码

1. this.props.childen(router/demo2)

/src/pages/router-demo/router2/router.js

```
1 import React from "react";
2 import { HashRouter as Router, Route } from "react-router-dom"; //导入路由, 为HashRouter起别名: Router
3 import About from "../../router1/About";
4 import Topic from "../../router1/Topic";
5 import Main from "../../router1/Main";
6 import Home from "../Home"; //4.0允许在路由中嵌套标签组件 这里是嵌套<Home>组件
7
8 export default class IRouter extends React.Component {
9   render() {
10     return (
11       <Router>
12         <Home>
13         <Route exact={true} path="/" component={Main} />
14         <Route path="/about" component={About} />
15         <Route path="/topics" component={Topic} />
16       </Home>
17     </Router>
18   );
19 }
20 }
```

• /src/pages/router-demo/router2/Home.js

```
1 import React from "react";
2 import { Link } from "react-router-dom"; //引入路由
3
4
5 export default class Home extends React.Component {
6   //默认输出 对象{}
7   render() {
8     return (
9       <div>
10         <ul>
11           <li>
12             <Link to="/">Home1</Link>
13           </li>
14           <li>
15             <Link to="/about">About1</Link>
16           </li>
17           <li>
18             <Link to="/topics">Topics1</Link>
19           </li>
20         </ul>
21       </div>
22     );
23   }
24 }
```

```

21     </ul>
22     <hr />
23     {this.props.children}
24   </div>
25 );
26 }
}

```

src\index.js

```

1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6 import Life from './pages/demo/Life';
7 import Admin from './admin';
8
9 // import Home from './pages/route_demo/router1/Home';
10 // ReactDOM.render(<Home />, document.getElementById('root'));
11
12
13
14 import Router from './pages/route_demo/router2/router';
15 ReactDOM.render(<Router />, document.getElementById('root'));
16
17
18 //import Router from './pages/route_demo/router3/router';
19 //ReactDOM.render(<Router />, document.getElementById('root'));
20
21
22 // ReactDOM.render(<Admin />, document.getElementById('root'));
23
24 // If you want your app to work offline and load faster, you can change
25 // unregister() to register() below. Note this comes with some pitfalls.
26 // Learn more about service workers: http://bit.ly/CRA-PWA
27 serviceWorker.unregister();

```

2.在子组件中嵌套路由(router/demo3)

src\pages\route_demo\router3\Home.js

```

1 import React from "react";
2 import { Link } from "react-router-dom"; //引入路由
3
4
5 export default class Home extends React.Component {
6   //默认输出 对象{}
7   render() {
8     return (
9       <div>
10         <ul>
11           <li>
12             <Link to="/main">Home1</Link>
13           </li>
14           <li>
15             <Link to="/about">About1</Link>
16           </li>
17           <li>
18             <Link to="/topics">Topics1</Link>
19           </li>
20         </ul>
21         <hr />
22         {this.props.children}
23       </div>
24     );
25   }
26 }

```

src\pages\route_demo\router3\Main.js

```

1 import React from 'react'
2 import {Link} from 'react-router-dom'
3 export default class Main extends React.Component {
4   render() {
5     return (
6       <div>
7         this is Main page.
8         <Link to = "/main/a">嵌套路由</Link>
9         <hr/>
10         {this.props.children}
11       </div>
12     );
13   }
14 }

```

src\pages\route_demo\router3\router.js

```
1 import React from "react";
2 import { HashRouter as Router, Route } from "react-router-dom"; //导入路由,为
3 HashRouter起别名: Router
4 import About from "../router1/About";
5 import Topic from "../router1/Topic";
6 import Main from "./Main";
7 import Home from "./Home"; //4.0允许在路由中嵌套标签组件 这里是嵌套<Home>组件
8
9 /* 嵌套路由中添加 render方法加载
10 在子组件中进行嵌套路由
11 */
12 export default class IRouter extends React.Component {
13   render() {
14     return (
15       <Router>
16         <Home>
17           <Route
18             path="/main"
19             render={() =>
20               <Main>
21                 { /* <div>this is a subchild element</div> */ }
22                 <Route path="/main/a" component={About} />
23               </Main>
24             )
25           </>
26           <Route path="/about" component={About} />
27           <Route path="/topics" component={Topic} />
28         </Home>
29       </Router>
30     );
31   }
32 }
```

src\index.js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6 import Life from './pages/demo/Life';
7 import Admin from './admin';
8
9 // import Home from './pages/route_demo/router1/Home';
10 // ReactDOM.render(<Home />, document.getElementById('root'));
11
12
13
14 // import Router from './pages/route_demo/router2/router';
15 // ReactDOM.render(<Router />, document.getElementById('root'));
16
17
18 import Router from './pages/route_demo/router3/router';
19 ReactDOM.render(<Router />, document.getElementById('root'));
20
21
22 // ReactDOM.render(<Admin />, document.getElementById('root'));
23
24 // If you want your app to work offline and load faster, you can change
25 // unregister() to register() below. Note this comes with some pitfalls.
26 // Learn more about service workers: http://bit.ly/CRA-PWA
27 serviceWorker.unregister();
```