

link: null
title: 珠峰架构师成长计划
description: publicindex.html
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=156 sentences=849, words=5003

内容大纲

- React路由原理
- 基本路由使用和实现
- 实现history
- 实现正则匹配
- 实现Switch
- 实现Redirect
- 实现Link
- 实现嵌套路由
- 实现受保护路由
- 实现NavLink
- 实现withRouter
- 实现Prompt
- 实现hooks

1. React路由原理

- 不同的路径渲染不同的组件
- 有两种实现方式
 - HashRouter: 利用hash实现路由切换
 - BrowserRouter: 实现h5 Api实现路由的切换

1.1 HashRouter

- 利用hash实现路由切换

publicindex.html

```
Document

#root{
  border:1px solid red;
}

/a
/b

window.addEventListener('hashchange', ()=>{
  console.log(window.location.hash);
  let pathname = window.location.hash.slice(1); //把最前面的那个#删除
  root.innerHTML = pathname;
});
```

1.2 BrowserRouter

- 利用h5 Api实现路由的切换

1.2.1 history

- HTML5规范给我们提供了一个[history \(https://developer.mozilla.org/zh-CN/docs/Web/API/Window/history\)](https://developer.mozilla.org/zh-CN/docs/Web/API/Window/history)接口
- HTML5 History API包括2个方法: history.pushState() 和 history.replaceState(), 和1个事件 window.onpopstate

1.2.1.1 pushState

- history.pushState(stateObject, title, url), 包括三个参数
 - 第一个参数用于存储该url对应的状态对象, 该对象可在onpopstate事件中获取, 也可在history对象中获取
 - 第二个参数是标题, 目前浏览器并未实现
 - 第三个参数则是设定的url
- pushState函数向浏览器的历史堆栈压入一个url为设定值的记录, 并改变历史堆栈的当前指针至栈顶

1.2.1.2 replaceState

- 该接口与pushState参数相同, 含义也相同
- 唯一的区别在于 replaceState是替换浏览器历史堆栈的当前历史记录为设定的url
- 需要注意的是 replaceState不会改动浏览器历史堆栈的当前指针

1.2.1.3 onpopstate

- 该事件是window的属性
- 该事件会在调用浏览器的前进、后退以及执行 history.forward、history.back、和 history.go触发, 因为这些操作有一个共性, 即修改了历史堆栈的当前指针
- 在不改变document的前提下, 一旦当前指针改变则会触发 onpopstate事件

1.2.1.4 案例

- 浏览器针对每个页面维护一个 History栈, 执行 pushState函数可压入设定的 url至栈顶, 同时修改当前指针
- 当执行 back和 forward操作时, history栈大小并不会改变 (history.length不变), 仅仅移动当前指针的位置
- 若当前指针在history栈的中间位置(非栈顶), 此时执行pushState会在指针当前的位置添加此条目, 并成为新的栈顶

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document title</title>
  <style>
    #root{
      border:1px solid red;
      height:20px;
    }
  </style>
</head>
<body>
  <div id="root">div>
    <script>
      var historyObj = window.history;

      window.onpopstate = (event) => {
        console.log(event.type, event.detail, state);
        window.scrollTo(0,0);
      }

      window.addEventListener("popstate", (event) => {
        console.log(event.type, event.state);
        window.scrollTo(0,0);
      });

      (function (historyObj) {
        let oldPopState = historyObj.popState;
        historyObj.popState = function (state, pathname) {
          let result = oldPopState.apply(historyObj, arguments);
          if (typeof window.onpopstate === "function") {
            window.onpopstate({state: state, detail: {pathname, state}});
          }
          return result;
        }
      })(historyObj);

      let oldHistoryLength = historyObj.length;
      window.addEventListener(() => {
        historyObj.popState({page: 1 | 1, title: 'page1' | 1, '/page1'});
        console.log(historyObj.length-oldHistoryLength);
      }, 1000);

      window.addEventListener(() => {
        historyObj.popState({page: 2 | 1, title: 'page2' | 1, '/page2'});
        console.log(historyObj.length-oldHistoryLength);
      }, 2000);

      window.addEventListener(() => {
        historyObj.popState({page: 3 | 1, title: 'page3' | 1, '/page3'});
        console.log(historyObj.length-oldHistoryLength);
      }, 3000);

      window.addEventListener(() => {
        historyObj.back();
        window.addEventListener(() => console.log(historyObj.length-oldHistoryLength), 100);
      }, 4000);

      window.addEventListener(() => {
        historyObj.popState({page: 5 | 1, title: 'page5' | 1, '/page5'});
        console.log(historyObj.length-oldHistoryLength);
      }, 5000);

      window.addEventListener(() => {
        historyObj.go(1);
        console.log(historyObj.length-oldHistoryLength);
      }, 6000);

    </script>
  </div>
</body>
</html>

```

2.基本路由

2.1 安装

```
npm i react-router-dom --save
```

2.2 src/index.js

```
src/index.js
```

```

import React from 'react';
import ReactDOM from 'react-dom';
import {HashRouter as Router, Route} from 'react-router-dom';
import Home from './components/Home';
import User from './components/User';
import Profile from './components/Profile';
ReactDOM.render(
  <Router>
    <div>
      <Route path="/" component={Home} exact/>
      <Route path="/user" component={User} />
      <Route path="/profile" component={Profile}/>
    </div>
  </Router>
, document.getElementById('root'));

```

2.3 Home.js

```
src/components/Home.js
```

```
import React, {Component} from 'react';
export default class Home extends React.Component{
  render() {
    console.log(this.props);
    return (
      <div>
        <p>Homep</p>
        <button onClick={() => this.props.history.push({pathname: '/user', state: {id: 1}})}>跳转到/userbutton</button>
      </div>
    )
  }
}
```

2.4 Home.js

src/components/User.js

```
import React, {Component} from 'react';
export default class User extends React.Component{
  render() {
    console.log(this.props.location.state);
    return (
      <div>
        <p>Userp</p>
        <button onClick={() => this.props.history.goBack()}>返回button</button>
      </div>
    )
  }
}
```

2.4 Profile.js

src/components/Profile.js

```
import React, {Component} from 'react';
export default class Profile extends Component{
  render() {
    return (
      <div>Profilediv</div>
    )
  }
}
```

3. 实现基本路由

3.1 src/index.js

- [react-router-dom/modules/index.js \(https://gitee.com/zhufengpeixun/react-router/blob/master/packages/react-router-dom/modules/index.js\)](https://gitee.com/zhufengpeixun/react-router/blob/master/packages/react-router-dom/modules/index.js)

```
import React from 'react';
import ReactDOM from 'react-dom';
+import {HashRouter as Router, Route} from './react-router-dom';
import Home from './components/Home';
import User from './components/User';
import Profile from './components/Profile';
ReactDOM.render(
  <Router>
    <Route path="/" component={Home} />
    <Route path="/user" component={User} />
    <Route path="/profile" component={Profile} />
  </Router>,
  document.getElementById('root')
);
```

3.2 RouterContext.js

src/react-router/RouterContext.js

```
import React from 'react'
export default React.createContext({});
```

3.3 Router.js

src/react-router/Router.js

```
import React from 'react'
import RouterContext from './RouterContext';
class Router extends React.Component{
  constructor(props) {
    super(props);
    this.state = {
      location: props.history.location
    }

    this.unlisten = props.history.listen((location) => {
      this.setState({location});
    });
  }
  componentWillUnmount() {
    this.unlisten && this.unlisten();
  }
  render() {
    let value = {
      location: this.state.location,
      history: this.props.history
    }
    return (
      <RouterContext.Provider value={value}>
        {this.props.children}
      </RouterContext.Provider>
    )
  }
}

export default Router;
```

3.4 Route.js

src/react-router/Route.js

```
import React from 'react'
import RouterContext from './RouterContext';
class Route extends React.Component{
  static contextType = RouterContext
  render() {
    const {history,location} = this.context;
    const {path,component:RouteComponent} = this.props;
    const match = location.pathname === path;
    let routeProps = {history,location};
    let element=null;
    if(match){
      element= <RouteComponent {...routeProps}/>
    }
    return element;
  }
}
export default Route;
```

3.5 react-router/index.js

src\react-router\index.js

```
export {default as Route} from './Route';
export {default as Router} from './Router';
export {default as RouterContext} from './RouterContext';
```

3.6 BrowserRouter.js

src\react-router-dom\BrowserRouter.js

```
import React from 'react'
import {Router} from '../react-router';
import {createBrowserHistory} from 'history';

class BrowserRouter extends React.Component{

  history = createBrowserHistory(this.props)
  render() {
    return (
      <Router history={this.history}>
        {this.props.children}
      </Router>
    )
  }
}
export default BrowserRouter;
```

3.7 HashRouter.js

src\react-router-dom\HashRouter.js

```
import React from 'react'
import {Router} from '../react-router';
import {createHashHistory} from 'history';
class HashRouter extends React.Component{
  history = createHashHistory()
  render() {
    return (
      <Router history={this.history}>
        {this.props.children}
      </Router>
    )
  }
}
export default HashRouter;
```

3.8 react-router-dom/index.js

src\react-router-dom\index.js

```
export * from '../react-router';
export {default as HashRouter} from './HashRouter';
export {default as BrowserRouter} from './BrowserRouter';
```

3.实现history

3.1 createBrowserHistory.js

src\history\createBrowserHistory.js

```

function createBrowserHistory() {
  const globalHistory = window.history;
  let listeners = [];
  let state;
  function listen(listener) {
    listeners.push(listener);
    return () => {
      listeners = listeners.filter(item => item !== listener);
    }
  }
  function go(n) {
    globalHistory.go(n);
  }
  window.addEventListener('popstate', () => {
    let location = {state: globalHistory.state, pathname: window.location.pathname};

    notify({action: "POP", location});
  });
  function goBack() {
    go(-1);
  }
  function goForward() {
    go(1);
  }
  function notify(newState) {
    Object.assign(history, newState);
    history.length = globalHistory.length;
    listeners.forEach(listener => listener(history.location));
  }
  function push(pathname, nextState) {
    const action = 'PUSH';
    if (typeof pathname === 'object') {
      state = pathname.state;
      pathname = pathname.pathname;
    } else {
      state = nextState;
    }
    globalHistory.pushState(state, null, pathname);
    let location = {state, pathname};
    notify({action, location});
  }
  const history = {
    action: 'POP',
    go,
    goBack,
    goForward,
    push,
    listen,
    location: {pathname: window.location.pathname, state: window.location.state}
  }
  return history;
}
export default createBrowserHistory;

```

3.2 createHashHistory.js

src/history/createHashHistory.js

```

function createHashHistory() {
  let stack = [];
  let index = -1;
  let action = 'POP';
  let state;
  let listeners = [];
  function listen(listener) {
    listeners.push(listener);
    return () => {
      listeners = listeners.filter(item => item !== listener);
    }
  }
  function go(n) {
    action = 'POP';
    index += n;
    let nextLocation = stack[index];
    state = nextLocation.state;
    window.location.hash = nextLocation.pathname;
  }
  let hashChangeHandler = () => {
    let pathname = window.location.hash.slice(1);
    Object.assign(history, {action, location: {pathname, state}});
    if (action === 'PUSH') {
      stack[++index] = history.location;
    }
    listeners.forEach(listener => listener(history.location));
  }
  function push(pathname, nextState) {
    action = 'PUSH';
    if (typeof pathname === 'object') {
      state = pathname.state;
      pathname = pathname.pathname;
    } else {
      state = nextState;
    }
    window.location.hash = pathname;
  }

  window.addEventListener('hashchange', hashChangeHandler);
  function goBack() {
    go(-1);
  }
  function goForward() {
    go(1);
  }
  const history = {
    action: 'POP',
    go,
    goBack,
    goForward,
    push,
    listen,
    location: {},
    location: {pathname: '/', state: undefined}
  }
  if (window.location.hash) {
    action = 'PUSH';
    hashChangeHandler();
  } else {
    window.location.hash = '/';
  }
  return history;
}
export default createHashHistory;

```

3.3 history/index.js

src/history/index.js

```

export {default as createBrowserHistory} from './createBrowserHistory';
export {default as createHashHistory} from './createHashHistory';

```

3.4 BrowserRouter.js

src/react-router-dom/BrowserRouter.js

```

import React from 'react'
import {Router} from '../react-router';
+import {createBrowserHistory} from '../history';
// 创建相应的历史对象，并且传入Router组件，原样渲染子组件
class BrowserRouter extends React.Component {
  // 不管是哪种创建历史对象的方法，得到的history 长的都一样，都像window.history
  history = createBrowserHistory(this.props) // window.history
  render() {
    return (
      <Router history={this.history}>
        {this.props.children}
      </Router>
    )
  }
}
export default BrowserRouter;

```

3.5 HashRouter.js

src/react-router-dom/HashRouter.js

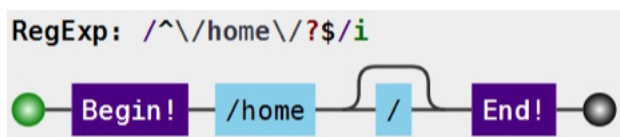
```
import React from 'react'
import {Router} from '../react-router';
import {createHashHistory} from '../history';
class HashRouter extends React.Component{
  history = createHashHistory()//hash实现
  render(){
    return (
      <Router history={this.history}>
        {this.props.children}
      </Router>
    )
  }
}
export default HashRouter;
```

4. path-to-regexp

- [regulex \(https://jex.im/regulex\)](https://jex.im/regulex)
- [path-to-regexp \(https://www.npmjs.com/package/path-to-regexp\)](https://www.npmjs.com/package/path-to-regexp)
 - `sensitive` 是否大小写敏感 (默认值: `false`)
 - `strict` 是否允许结尾有一个可选的/ (默认值: `false`)
 - `end` 是否匹配整个字符串 (默认值: `true`)

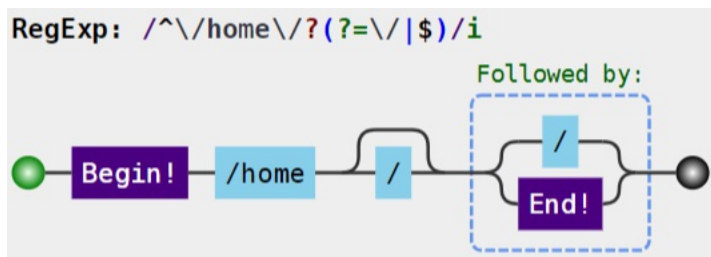
4.1 /home结束

```
let pathToRegExp = require('path-to-regexp');
let regexp = pathToRegExp('/home', [], {end:true});
console.log(regexp);
console.log(regexp.test('/home'));
console.log(regexp.test('/home/2'));
```



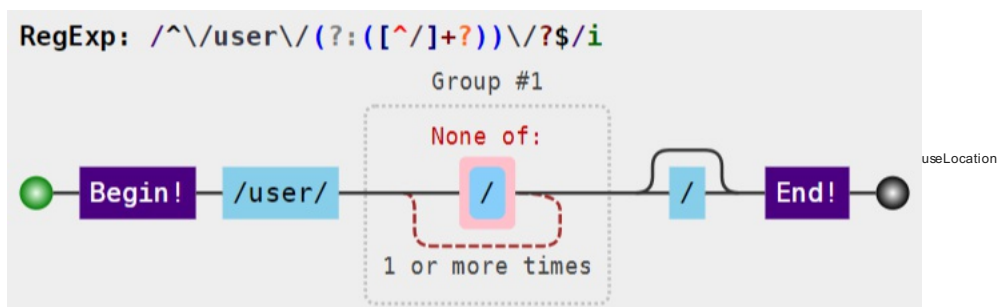
4.2 /home非结束

```
let pathToRegExp = require('path-to-regexp');
let regexp = pathToRegExp('/home', [], {end:false});
console.log(regexp);
console.log(regexp.test('/home'));
console.log(regexp.test('/home/'));
console.log(regexp.test('/home/'));
console.log(regexp.test('/home/2'));
```



4.3 路径参数

```
let params = [];
let regexp = pathToRegExp('/user/:id', params, {end:true});
console.log(regexp, params);
```



4.4 正则匹配

- 是否捕获

表达式 含义 `()` 表示捕获分组, `()` 会把每个分组里的匹配的值保存起来, 使用 `$n` (`n` 是一个数字, 表示第 `n` 个捕获组的内容) `(?:)` 表示非捕获分组, 和捕获分组唯一的区别在于, 非捕获分组匹配的值不会保存起来 (`<?>` `<name>...` `</name>`)

表示命名捕获分组, 反向引用一个命名分组的语法是 `<k<name></name>`

, 在

`replace()` `方` `法` `的` `替` `换` `字` `符` `串` `中` `反` `向` `引` `用` `是` `用`

```
console.log('1ab'.match(/1[a-z]([b-c])/));

console.log('1ab'.match(/1[a-z](?:[a-z])/));

console.log(/(?:\d{2})-(?:\d{2})/.exec('11-22'));
console.log('11-22'.replace(/(?:\d{2})-(?:\d{2})/, "$-{{content}}quot;));
```

- 前瞻和后顾

表达式 含义 (?:=pattern) 正向肯定查找(前瞻),后面必须跟着什么 (?!pattern) 正向否定查找(前瞻), 后面不能跟着什么 ``(?:<=pattern)< code><p class="node-read-div2p"> 反向肯定条件查找(后顾),不捕获 </p><code>(?!pattern)</code><p class="node-read-div2p"> 反向否定条件查找 (后顾) </p><pre><code class="lang-js">

```
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">'1a'</span>.match(<span class="hljs-regexp">/d(?:=[a-z])/</span>));
```

```
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">'1a'</span>.match(<span class="hljs-regexp">/d(?:[A-Z])/</span>));
```

```
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">'1a'</span>.match(<span class="hljs-regexp">/(?:=[a-z])d[a-z] < span>));
```

```
<span class="hljs-built_in">console</span>.log(<span class="hljs-string">'1a'</span>.match(<span class="hljs-regexp">/(?:<![A-Z])d[a-z]/</span>));
```

```
<span class="hljs-keyword">let</span><span class="hljs-string">"1ab"</span>;
```

```
array.index = <span class="hljs-number">0</span>;
```

```
array.input = <span class="hljs-string">"1ab"</span>;
```

```
array.groups = <span class="hljs-literal">undefined</span>;
```

```
<span class="hljs-built_in">console</span>.log(array);
```

```
</span><![a-z])d[a-z]/</span></code></pre><h2 id="t355. 正则匹配">5. 正则匹配 <a href="#t355. 正则匹配">#</a></h2><h3 id="t365.1 matchPath.js">5.1 matchPath.js <a href="#t365.1 matchPath.js">#</a></h3><p>src/react-router/matchPath.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span><span class="hljs-keyword"> pathToRegexp </span><span class="hljs-keyword">from</span><span class="hljs-keyword"> <span class="hljs-string">"path-to-regexp"</span>;
```

```
<span class="hljs-keyword">function</span><span class="hljs-keyword"> compilePath</span><span>(<span class="hljs-params">path,options</span></span>){
```

```
<span class="hljs-keyword">const</span> keys = [];
```

```
<span class="hljs-keyword">const</span> regexp = pathToRegexp(path,keys,options);
```

```
<span class="hljs-keyword">return</span> {keys,regexp};
```

```
}
```

```
<span class="hljs-function"><span class="hljs-keyword">function</span></span> <span class="hljs-title">matchPath</span><span>(<span class="hljs-params">pathname,options = {}</span></span>){
```

```
<span class="hljs-keyword">let</span> {path=<span class="hljs-string">'/'</span>,exact=<span class="hljs-literal">>false</span>,strict=<span class="hljs-literal">>false</span>,sensitive=<span class="hljs-literal">>false</span>}=options;
```

```
<span class="hljs-keyword">let</span> {keys,regexp} = compilePath(path,{
```

```
<span class="hljs-attr">end</span>:exact,
```

```
strict,
```

```
sensitive
```

```
});
```

```
<span class="hljs-keyword">const</span> match = regexp.exec(pathname);
```

```
<span class="hljs-keyword">if</span>(!match) <span class="hljs-keyword">return</span> <span class="hljs-literal">>null</span>;
```

```
<span class="hljs-keyword">const</span> [url,...values] = match;
```

```
<span class="hljs-keyword">const</span> isExact = pathname === url;
```

```
<span class="hljs-keyword">if</span>(exact && !isExact) <span class="hljs-keyword">return</span> <span class="hljs-literal">>null</span>;
```

```
<span class="hljs-keyword">return</span> {
```

```
path,
```

```
url,
```

```
isExact,
```

```
<span class="hljs-attr">params</span>:keys.reduce(<span class="hljs-function"><span class="hljs-params">memo,key,index</span></span>=></span>{
```

```
memo[key.name] = values[index];
```

```
<span class="hljs-keyword">return</span> memo;
```

```
},{})
```

```
}
```

```
}
```

```
<span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> matchPath;
```

```
</code></pre><h3 id="t375.2 Router.js">5.2 Router.js <a href="#t375.2 Router.js">#</a></h3><p>src/react-router/Router.js</p><pre><code class="lang-diff">import React from 'react'
import RouterContext from './RouterContext';
```

```
class Router extends React.Component{
```

```
<span class="hljs-addition">+ static computeRootMatch(pathname) {</span>
```

```
<span class="hljs-addition">+ return { path: "/", url: "/", params: {}, isExact: pathname === "/" };</span>
```

```
<span class="hljs-addition">+ }</span>
```

```
constructor(props){
```

```
super(props);
```

```
this.state = {
```

```
location:props.history.location
```

```
}
```

```
//当路径发生变化的时候执行回调
```

```
this.unlisten = props.history.listen((location)=>{
```

```
this.setState({location});
```

```
});
```

```
}
```

```
componentWillUnmount(){
```

```
this.unlisten&&this.unlisten();
```

```
}
```

```
render(){
```

```
let value = {/通过value向下层传递数据
```

```
location:this.state.location,
```

```
history:this.props.history,
```

```
<span class="hljs-addition">+ match: Router.computeRootMatch(this.state.location.pathname)</span>
```

```
}
```

```
return (
```

```
<routercontext.provider value={value}>
```

```
{this.props.children}
```

```
</routercontext.provider>
```

```
)
```

```
}
```

```
}
```

```
export default Router;
```

```
</code></pre><h3 id="t385.3 Route.js">5.3 Route.js <a href="#t385.3 Route.js">#</a></h3><p>src/react-router/Route.js</p><pre><code class="lang-diff">import React from 'react'
```

```
import RouterContext from './RouterContext';
```

```
<span class="hljs-addition">+import matchPath from './matchPath';</span>
```

```
class Route extends React.Component{
```

```
static contextType = RouterContext
```

```
render(){
```

```
const {history,location} = this.context;
```

```
const {component:RouteComponent} = this.props;
```

```
<span class="hljs-addition">+ const match = matchPath(location.pathname, this.props);</span>
```

```
let routeProps = {history,location};
```



```

let element=null;
if (match) {
  <span class="hljs-addition">+ routeProps.match = match;</span>
  element = <routeComponent {...routeProps}>
}
return (
  <routerContext.provider value="{routeProps}">
    {element}
  </routerContext.provider>
)
}
}
export default Route;
</routeComponent></code></pre><h3 id="t395.4 react-router\index.js">5.4 react-router\index.js<a href="#t395.4 react-router\index.js">#</a></h3><p>src\react-router\index.js</p><pre><code class="lang-diff">export {default as Route} from './Route';
export {default as Router} from './Router';
export {default as __RouterContext} from './RouterContext';
<span class="hljs-addition">+export {default as matchPath} from './matchPath';</span>
</code></pre><h2 id="t406. 实现Switch">6. 实现Switch<a href="#t406. 实现Switch">#</a></h2><h3 id="t416.1 Switch.js">6.1 Switch.js<a href="#t416.1 Switch.js">#</a></h3><p>src\react-router\Switch.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span></span> React <span class="hljs-keyword">from</span></span> <span class="hljs-string">'react'</span>
<span class="hljs-keyword">import</span></span></span> RouterContext <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./RouterContext'</span></span>;
<span class="hljs-keyword">import</span></span></span> matchPath <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./matchPath'</span></span>;
<span class="hljs-class"><span class="hljs-class"><span class="hljs-keyword">class</span></span> Route <span class="hljs-keyword">extends</span></span> React</span></span> {
  <span class="hljs-keyword">static</span></span> contextType = RouterContext
  render() {
    <span class="hljs-keyword">const</span></span> { location } = <span class="hljs-keyword">this</span></span>.context;
    <span class="hljs-keyword">let</span></span> element, match;
    React.Children.forEach(<span class="hljs-keyword">this</span></span>.props.children, child => {
      <span class="hljs-keyword">if</span></span> (<span class="hljs-keyword">match</span></span> && React.isValidElement(child)) {
        element = child;
        match = matchPath(location.pathname, child.props);
      }
    });
    <span class="hljs-keyword">return</span></span> match ? React.cloneElement(element, {<span class="hljs-attr">computedMatch</span></span>: match }) : <span class="hljs-literal">null</span></span>
  }
}
<span class="hljs-keyword">export</span></span></span> <span class="hljs-keyword">default</span></span></span> Switch;
</code></pre><h3 id="t426.2 Route.js">6.2 Route.js<a href="#t426.2 Route.js">#</a></h3><p>src\react-router\Route.js</p><pre><code class="lang-diff">import React from 'react'
import RouterContext from './RouterContext';
import matchPath from './matchPath';
class Route extends React.Component{
  static contextType = RouterContext
  render(){
    const {history,location} = this.context;
    <span class="hljs-addition">+ const {component:RouteComponent,computedMatch} = this.props;</span>
    <span class="hljs-addition">+ const match = computedMatch ? computedMatch : matchPath(location.pathname, this.props);</span>
    let routeProps = {history,location};
    let element=null;
    if (match) {
      routeProps.match = match;
      element = <routeComponent {...routeProps}>
    }
    return (
      <routerContext.provider value="{routeProps}">
        {element}
      </routerContext.provider>
    )
  }
}
export default Route;
</routeComponent></code></pre><h3 id="t436.3 react-router\index.js">6.3 react-router\index.js<a href="#t436.3 react-router\index.js">#</a></h3><p>src\react-router\index.js</p><pre><code class="lang-diff">export {default as Route} from './Route';
export {default as Router} from './Router';
export {default as __RouterContext} from './RouterContext';
export {default as matchPath} from './matchPath';
<span class="hljs-addition">+export {default as Switch} from './Switch';</span>
</code></pre><h3 id="t446.4 src\index.js">6.4 src\index.js<a href="#t446.4 src\index.js">#</a></h3><p>src\index.js</p><pre><code class="lang-diff">import React from 'react';
import ReactDOM from 'react-dom';
import {HashRouter as Router,Route,Switch} from './react-router-dom';
import Home from './components/Home';
import User from './components/User';
import Profile from './components/Profile';
ReactDOM.render(
  <router>
    <span class="hljs-addition">+ <switch></switch></span>
    <route path="/" component="{Home}" exact>
    <route path="/user" component="{User}">
    <route path="/profile" component="{Profile}">
    <span class="hljs-addition">+ </span>
  </route></route></route></router>
  ,document.getElementById('root'));
</code></pre><h2 id="t457. Redirect">7. Redirect<a href="#t457. Redirect">#</a></h2><h3 id="t467.1 Redirect.js">7.1 Redirect.js<a href="#t467.1 Redirect.js">#</a></h3><p>src\react-router\Redirect.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span></span> React <span class="hljs-keyword">from</span></span> <span class="hljs-string">'react'</span>
<span class="hljs-keyword">import</span></span></span> RouterContext <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./RouterContext'</span></span>;
<span class="hljs-keyword">import</span></span></span> Lifecycle <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./Lifecycle'</span></span>;

<span class="hljs-function"><span class="hljs-keyword">function</span></span> Redirect</span>(<span class="hljs-params">{to}</span></span>){
  <span class="hljs-keyword">return</span></span> (
    <span class="hljs-tag"><span class="hljs-tag"><span class="hljs-name">RouterContext.Consumer</span></span></span>
    {
      contextValue=>{
        const {history}= contextValue;
        return (
          <span class="hljs-tag"><span class="hljs-tag"><span class="hljs-name">Lifecycle</span></span>
          <span class="hljs-attr">onMount</span>=<span class="hljs-string">{()=>=</span>=</span>history.push(to)}
        )
      }
    }
  )
}
<span class="hljs-tag"><span class="hljs-tag"><span class="hljs-name">RouterContext.Consumer</span></span></span>
);
}
<span class="hljs-keyword">export</span></span></span> <span class="hljs-keyword">default</span></span></span> Redirect;

```

[illegible]

```


```

```

import Home from './components/Home';
import User from './components/User';
import Profile from './components/Profile';
<span class="hljs-addition">+import Protected from './components/Protected';</span>
<span class="hljs-addition">+import Login from './components/Login';</span>
ReactDOM.render(
  <router>
    <ul>
      <li><link to="/"> 首页</li>
      <li><link to="/user">用户管理</li>
      <li><link to="/profile">个人中心</li>
    </ul>
    <switch>
      <route path="/" component="(Home)" exact>
      <route path="/user" component="(User)">
      <span class="hljs-addition">+ <protected path="/profile" component="(Profile)"/></protected></span>
      <span class="hljs-addition">+ <route path="/login" component="(Login)"/></route></span>
      <redirect to="/">
    </redirect></route></switch>
  </router>
  ,document.getElementById('root'));
</code></pre><h3 id="t6210.2 Route.js">10.2 Route.js <a href="#t6210.2 Route.js">#</a></h3><p>src/react-router</p><pre><code class="lang-diff">import React from 'react'
import RouterContext from './RouterContext';
import matchPath from './matchPath';
class Route extends React.Component{
  static contextType = RouterContext
  render(){
    const {history,location} = this.context;
    <span class="hljs-addition">+ const {component:RouteComponent,computedMatch,render} = this.props;</span>
    const match = computedMatch ? computedMatch : matchPath(location.pathname, this.props);
    let routeProps = {history,location};
    let element=null;
    <span class="hljs-addition">+ if (match) {</span>
    <span class="hljs-addition">+ routeProps.match = match;</span>
    <span class="hljs-addition">+ if (RouteComponent) {</span>
    <span class="hljs-addition">+ element = <routecomponent {...routeprops}></routecomponent></span>
    <span class="hljs-addition">+ } else if (render) {</span>
    <span class="hljs-addition">+ element= render(routeProps);</span>
    <span class="hljs-addition">+ } else {</span>
    <span class="hljs-addition">+ element= null;</span>
    <span class="hljs-addition">+ }</span>
    <span class="hljs-addition">+ }else {</span>
    <span class="hljs-addition">+ element= null;</span>
    <span class="hljs-addition">+ }</span>
    return (
    <routercontext.provider value="{routeProps}">
      (element)
    </routercontext.provider>
    )
  }
}
export default Route;
</code></pre><h3 id="t6310.3 Login.js">10.3 Login.js <a href="#t6310.3 Login.js">#</a></h3><p>src/components/Login.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span> </span> React
<span class="hljs-keyword">from</span> </span> './components/Login';
<span class="hljs-keyword">import</span> </span> Protected <span class="hljs-keyword">from</span> </span> './components/Protected';
<span class="hljs-keyword">import</span> </span> Redirect <span class="hljs-keyword">from</span> </span> './components/Redirect';
<span class="hljs-keyword">import</span> </span> matchPath <span class="hljs-keyword">from</span> </span> './matchPath';
class Login extends React.Component{
  static contextType = RouterContext
  render(){
    const {history,location} = this.context;
    const {login} = this.props;
    const match = matchPath(location.pathname, this.props);
    let element=null;
    <span class="hljs-addition">+ if (match) {</span>
    <span class="hljs-addition">+ element = <login>{...this.props,match}</span>
    <span class="hljs-addition">+ } else {</span>
    <span class="hljs-addition">+ element = null;</span>
    <span class="hljs-addition">+ }</span>
    return (
    <routercontext.provider value="{login}">
      (element)
    </routercontext.provider>
    )
  }
}
export default Login;
</code></pre><h3 id="t6410.4 Protected.js">10.4 Protected.js <a href="#t6410.4 Protected.js">#</a></h3><p>src/components/Protected.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span> </span> React
<span class="hljs-keyword">from</span> </span> './components/Protected';
<span class="hljs-keyword">import</span> </span> Redirect <span class="hljs-keyword">from</span> </span> './components/Redirect';
<span class="hljs-keyword">import</span> </span> matchPath <span class="hljs-keyword">from</span> </span> './matchPath';
class Protected extends React.Component{
  static contextType = RouterContext
  render(){
    const {history,location} = this.context;
    const {login} = this.props;
    const match = matchPath(location.pathname, this.props);
    let element=null;
    <span class="hljs-addition">+ if (match) {</span>
    <span class="hljs-addition">+ element = <login>{...this.props,match}</span>
    <span class="hljs-addition">+ } else {</span>
    <span class="hljs-addition">+ element = null;</span>
    <span class="hljs-addition">+ }</span>
    return (
    <routercontext.provider value="{login}">
      (element)
    </routercontext.provider>
    )
  }
}
export default Protected;
</code></pre><h3 id="t6611.1 publicindex.html">11.1 publicindex.html <a href="#t6611.1 publicindex.html">#</a></h3><p>publicindex.html</p><pre><code class="lang-diff"><!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="theme-color" content="#000000">
  <meta name="description" content="Web site created using create-react-app">
  <title>React App</title>
  <style>
    <span class="hljs-addition">+ .strong{</span>
    <span class="hljs-addition">+ font-size: 20px;</span>
    <span class="hljs-addition">+ }</span>
    <span class="hljs-addition">+ .active{</span>
    <span class="hljs-addition">+ background-color: green;</span>
    <span class="hljs-addition">+ }</span>
  </style>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition">+ <background-color: green></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <strong></span>
  <span class="hljs-addition">+ <font-size: 20px></span>
  <span class="hljs-addition">+ </span>
  <span class="hljs-addition">+ <active></span>
  <span class="hljs-addition
```

```
<style>  
</head>  
<body>  
  
</body>  
</html>  
</code></pre><h3 id="t6711.2 src\index.js">11.2 src\index.js <a href="#t6711.2 src\index.js">#</a></h3><p>src\index.js</p><pre><code class="lang-diff">import React from 'react';  
import ReactDOM from 'react-dom';  
<span class="hljs-addition">+import {HashRouter as Router,Route,Switch,Redirect,NavLink} from './react-router-dom';</span>  
import Home from './components/Home';  
import User from './components/User';  
import Profile from './components/Profile';  
import Protected from './components/Protected';  
import Login from './components/Login';  
ReactDOM.render(  
<router>  
<ul>  
<span class="hljs-addition">+<i><navlink classname="strong" style="{textDecoration:'line-through'} activeStyle="{color:red}" to="/" exact=Home</navlink></li></span>  
<span class="hljs-addition">+<i><navlink activeStyle="{color:red}" to="/user">User</navlink></li></span>  
<span class="hljs-addition">+<i><navlink activeStyle="{color:red}" to="/profile">Profile</navlink></li></span>  
</ul>  
<switch>  
<route path="/" component="(Home)" exact>  
<route path="/user" component="(User)">  
<protected path="/profile" component="(Profile)">  
<route path="/login" component="(Login)">  
<redirect to="/">  
</redirect></route></protected></route></route></switch>  
</router>  
,document.getElemenById('root'));  
</code></pre><p>src/react-router-dom\NavLink.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-keyword">React</span>  
string">'react'</span>;  
<span class="hljs-keyword">import</span> <span class="hljs-keyword">(Route)</span> <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./react-router'</span>;  
<span class="hljs-keyword">import</span> <span class="hljs-keyword">matchPath</span> <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./react-router/matchPath'</span>;  
<span class="hljs-keyword">import</span> <span class="hljs-keyword">Link</span> <span class="hljs-keyword">from</span></span> <span class="hljs-string">'./Link'</span>;  
<span class="hljs-function"><span class="hljs-keyword">function</span></span> <span class="hljs-keyword">function</span> <span class="hljs-title">NavLink</span></span><span class="hljs-params"><span class="hljs-keyword">props</span></span></span>{  
<span class="hljs-keyword">const</span> <span class="hljs-keyword">{</span>  
<span class="hljs-attr">to</span></span>:<span class="hljs-attr">path,</span>  
<span class="hljs-attr">className</span></span>:<span class="hljs-attr">classNameProp</span><span class="hljs-string">"</span></span>,<br><span class="hljs-attr">style</span></span>:<span class="hljs-attr">styleProp</span>=<span class="hljs-attr">activeClassName</span>=<span class="hljs-string">"active"</span>,<br>activeStyle=<span class="hljs-attr">children,</span><br>exact<br>}=<span class="hljs-keyword">props;</span>  
<span class="hljs-keyword">return</span></span> (<br><span class="hljs-keyword">xml"><span class="hljs-tag"><span class="hljs-name">Route</span> <span class="hljs-attr">path</span>=<span class="hljs-string">{path}</span> <span class="hljs-attr">exact</span>=<span class="hljs-string">{exact}</span></span></span>  
{<br>((match))=>{<br>let className = match?joinClassNames(classNameProp,activeClassName):classNameProp;<br>let style =match?(...styleProp,...activeStyle):styleProp;<br>let linkProps = {<br>  className,<br>  style,<br>  to:path,<br>  children<br>}<br>return <span class="hljs-tag"><span class="hljs-name">Link</span> <span class="hljs-attr">...linkProps</span></span></span><br>}<br><span class="hljs-tag"><span class="hljs-name">Route</span></span></span>  
)<br>}<br><span class="hljs-function"><span class="hljs-keyword">function</span></span> <span class="hljs-title">NavLink</span></span><span class="hljs-params"><span class="hljs-keyword">props</span></span></span>{<br><span class="hljs-keyword">let</span></span> context = React.useContext(RouterContext);<br><span class="hljs-keyword">let</span></span> <span class="hljs-attr">location</span>=<span class="hljs-attr">pathname}</span>= context;<br><span class="hljs-keyword">const</span></span> {<br><span class="hljs-attr">to</span></span>:<span class="hljs-attr">path,</span><br><span class="hljs-attr">className</span></span>:<span class="hljs-attr">classNameProp</span><span class="hljs-string">"</span></span>,<br><span class="hljs-attr">style</span></span>:<span class="hljs-attr">styleProp</span>=<span class="hljs-attr">activeClassName</span>=<span class="hljs-string">"active"</span>,<br>activeStyle=<span class="hljs-attr">children,</span><br>exact<br>}=<span class="hljs-keyword">props;</span>  
<span class="hljs-keyword">let</span></span> isActive = matchPath(pathname,{path,exact});<br><span class="hljs-keyword">let</span></span> className = isActive?joinClassNames(classNameProp,activeClassName):classNameProp;<br><span class="hljs-keyword">let</span></span> style = isActive?(...styleProp,...activeStyle):styleProp;<br><span class="hljs-keyword">let</span></span> linkProps = {<br>  className,<br>  style,<br>  <span class="hljs-attr">to</span></span>:<span class="hljs-attr">path,</span><br>  children<br>}<br><span class="hljs-keyword">return</span></span> <span class="hljs-keyword">xml"><span class="hljs-tag"><span class="hljs-name">Link</span> <span class="hljs-attr">...linkProps</span></span></span></span></code></pre><h2 id="t6911.4 react-router-dom\index.js">11.4 react-router-dom\index.js <a href="#t6911.4 react-router-dom\index.js">#</a></h3><p>src\react-router-dom\index.js</p><pre><code class="lang-diff">export * from './react-router';  
export {default as HashRouter} from './HashRouter';  
export {default as BrowserRouter} from './BrowserRouter';  
export {default as Link} from './Link';  
<span class="hljs-addition">+export { default as NavLink } from './NavLink';</span>  
</code></pre><h2 id="t7012.withRouter">#</a></h2><h3 id="t7112.1 src\index.js">12.1 src\index.js <a href="#t7112.1 src\index.js">#</a></h3><p>src\index.js</p><pre><code class="lang-diff">import React from 'react';  
import ReactDOM from 'react-dom';
```

```
import {HashRouter as Router,Route,Switch,Redirect,NavLink} from './react-router-dom';
import Home from './components/Home';
import User from './components/User';
import Profile from './components/Profile';
import Protected from './components/Protected';
import Login from './components/Login';
<span class="hljs-addition">+import NavHeader from './components/NavHeader';</span>
ReactDOM.render(
  <router>
    <>
      <span class="hljs-addition">+ <navheader title="欢迎光临"></navheader></span>
      <ul>
        <li><navlink classname="strong" style="{(textDecoration:" 'line-through')}" activestyle="{({color:'red'})}" to="/" exact=Home</navlink></li>
        <li><navlink activestyle="{({color:'red'})}" to="/user">User</navlink></li>
        <li><navlink activestyle="{({color:'red'})}" to="/profile">Profile</navlink></li>
      </ul>
      <switch>
        <route path="/" component="(Home)" exact>
          <route path="/user" component="(User)">
            <protected path="/profile" component="(Profile)"/>
            <route path="/login" component="(Login)"/>
            <redirect to="/">
          </redirect></route></protected></route></route></switch>
        </>
      </router>
      ,document.createElementByld('root'));
    </code></pre><p>src\components\NavHeader.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-string">'./react-router-dom'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {withRouter} <span class="hljs-keyword">from</span> <span class="hljs-string">'./react-router-dom'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> RouterContext <span class="hljs-keyword">from</span> <span class="hljs-string">'./RouterContext'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {OldComponent} <span class="hljs-keyword">from</span> <span class="hljs-string">'./OldComponent'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {RouterContext.Consumer} <span class="hljs-keyword">from</span> <span class="hljs-string">'./RouterContext.Consumer'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {Prompt} <span class="hljs-keyword">from</span> <span class="hljs-string">'./Prompt'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {useNameRef} <span class="hljs-keyword">from</span> <span class="hljs-string">'./utils'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {useState} <span class="hljs-keyword">from</span> <span class="hljs-string">'react'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {ReactDOM} <span class="hljs-keyword">from</span> <span class="hljs-string">'react-dom'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> {BrowserRouter as Router,Route,Switch,Redirect,NavLink} <span class="hljs-keyword">from</span> <span class="hljs-string">'./react-router-dom'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> Home <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/Home'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> User <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/User'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> Profile <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/Profile'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> Protected <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/Protected'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> Login <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/Login'</span>;
<span class="hljs-class"><span class="hljs-keyword">import</span></span> NavHeader <span class="hljs-keyword">from</span> <span class="hljs-string">'./components/NavHeader'</span>;
ReactDOM.render(
  <span class="hljs-addition">+ <router getuserconfirmation="(())=>{return window.confirm()}</router></span>
  <>
    <navheader title="欢迎光临">
      <ul>
        <li><navlink classname="strong" style="{(textDecoration:" 'line-through')}" activestyle="{({color:'red'})}" to="/" exact=Home</navlink></li>
        <li><navlink activestyle="{({color:'red'})}" to="/user">User</navlink></li>
        <li><navlink activestyle="{({color:'red'})}" to="/profile">Profile</navlink></li>
      </ul>
      <switch>
        <route path="/" component="(Home)" exact>
          <route path="/user" component="(User)">
            <protected path="/profile" component="(Profile)"/>
            <route path="/login" component="(Login)"/>
            <redirect to="/">
          </redirect></route></protected></route></route></switch>
        </>
      </>
    </>
    ,document.createElementByld('root'));
  </navheader></code></pre><h3 id="t7713.2 UserAdd.js">13.2 UserAdd.js</h3><p>src\components\UserAdd.js</p><pre><code class="lang-diff">import React, {
  Component } <span class="hljs-keyword">from</span> 'react';
import {UserAPI} <span class="hljs-keyword">from</span> './utils';
import {Prompt} <span class="hljs-keyword">from</span> './react-router-dom';
export default class UserAdd extends Component {
  usernameRef
  <span class="hljs-addition">+ state = ({isLoading:false})</span>
  constructor(props) {
    super(props);
    </code>
```

```
this.usernameRef = React.createRef();
}
handleSubmit = (event) => {
  event.preventDefault();
  <span class="hljs-addition">+ this.setState({</span>
  <span class="hljs-addition">+ isBlocking:false</span>
  <span class="hljs-addition">+ },{}</span>
  let username = this.usernameRef.current.value;
  UserAPI.add({ id: Date.now() + "", username });
  this.props.history.push('/user/list');
  <span class="hljs-addition">+ });</span>
}
render() {
  return (
    <form onSubmit={this.handleSubmit}>
      <span class="hljs-addition">+ <prompt< span>
      <span class="hljs-addition">+ when={this.state.isBlocking}</span>
      <span class="hljs-addition">+ message = {</span>
      <span class="hljs-addition">+ (location)=&#x8BF7; &#x95EE; &#x4F60; &#x786E; &#x5B9A; &#x8981; &#x8DF3; &#x8F6C; &#x5230; ${location.pathname} &#x5417; ?</span>
      <span class="hljs-addition">+ }</span>
      <span class="hljs-addition">+ /></span>
      <span class="hljs-addition">+ <input type="text" ref={this.usernameRef} onChange={(event) => {</span>
      <span class="hljs-addition">+ this.setState({ isBlocking: event.target.value.length > 0 });</span>
      <span class="hljs-addition">+ } } /></span>
      <button type="submit">提交</button>
    </prompt></span></form>
  )
}
}
}
</code></pre><h3 id="t7813.3 Prompt.js">13.3 Prompt.js <a href="#t7813.3 Prompt.js">#</a></h3><p>src/react-router/Prompt.js</p><pre><code class="lang-js"><span class="hljs-keyword">import</span>
React <span class="hljs-keyword">from</span> <span class="hljs-string">'react'</span>;
<span class="hljs-keyword">import</span> <span class="hljs-keyword">RouterContext</span> <span class="hljs-keyword">from</span> <span class="hljs-string">'./RouterContext'</span>;
<span class="hljs-keyword">import</span> <span class="hljs-keyword">Lifecycle</span> <span class="hljs-keyword">from</span> <span class="hljs-string">'./Lifecycle'</span>;

<span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">Prompt</span>(<span class="hljs-params">props</span>)</span>{
  <span class="hljs-keyword">let</span> <span class="hljs-keyword">value</span> = React.useContext(RouterContext);
  React.useEffect(<span class="hljs-function"><span class="hljs-keyword">()</span></span> <span class="hljs-params">()</span></span>=>{
    <span class="hljs-keyword">return</span> <span class="hljs-keyword">value</span>.history.block(props.message)
  });
  <span class="hljs-keyword">return</span> <span class="hljs-keyword">null</span>;
}

<span class="hljs-class"><span class="hljs-keyword">class</span> <span class="hljs-keyword">Component</span> {
  <span class="hljs-keyword">static</span> contextType = RouterContext;
  componentDidMount(){
    <span class="hljs-keyword">const</span> <span class="hljs-keyword">block</span> = <span class="hljs-keyword">this</span>.context.history.block;
    <span class="hljs-keyword">this</span>.release = <span class="hljs-keyword">block</span>(<span class="hljs-keyword">this</span>.props.message)
  }
  componentWillUnmount(){
    <span class="hljs-keyword">this</span>.release()
  }
  render(){
    <span class="hljs-keyword">return</span> <span class="hljs-keyword">null</span>;
  }
}

<span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">Prompt</span>(<span class="hljs-params">{when,message}</span>)</span>{
  <span class="hljs-keyword">return</span> (
    <span class="hljs-tag"><span class="hljs-name">RouterContext.Consumer</span></span>
  ) {
    (value)=>{
      //如果不需要阻止跳转，则可以直接返回null,什么都不做
      if(!when) return null;
      //需要给history对象上添加一个block方法
      const block = value.history.block;
      return (
        <span class="hljs-tag"><span class="hljs-name">Lifecycle</span>
        <span class="hljs-attr">onMount</span>=<span class="hljs-string">{lifeCycleInstance</span>=><span>lifeCycleInstance.release = block(message)}
        onUnmount={lifeCycleInstance=>lifeCycleInstance.release()}
      )
    }
  }
}

<span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> Prompt;
</code></pre><h3 id="t7913.4 react-router/index.js">13.4 react-router/index.js <a href="#t7913.4 react-router/index.js">#</a></h3><p>src/react-router/index.js</p><pre><code class="lang-diff">export
{default as Route} from './Route';
export {default as Router} from './Router';
export {default as __RouterContext} from './RouterContext';
export {default as matchPath} from './matchPath';
export {default as Switch} from './Switch';
export {default as Redirect} from './Redirect';
export {default as withRouter} from './withRouter';
<span class="hljs-addition">+ export {default as Prompt} from './Prompt';</span>
</code></pre><h3 id="t8013.5 createBrowserHistory.js">13.5 createBrowserHistory.js <a href="#t8013.5 createBrowserHistory.js">#</a></h3><p>src/history/createBrowserHistory.js</p><pre><code class="lang-diff">function createBrowserHistory(props){
  <span class="hljs-addition">+ let confirm = props.getUserConfirmation?props.getUserConfirmation():window.confirm;</span>
  let globalHistory = window.history;
  let listeners = [];
  <span class="hljs-addition">+ let message;</span>
  function go(n){
    globalHistory.go(n);
  }
  function goBack(){
    go(-1)
  }
  function goForward(){
    go(1)
  }
}
```

[illegible]

[illegible]