link null title: 珠峰架构师成长计划 description: public\index.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
- ◆ 空型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
- 主型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

public\index.html

```
Document
     #root{
          border:1px solid red;
     /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規範給我们提供了一个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

- 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项

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
- ullet 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至栈项,同时修改当前指针
 当执行 back和 forward操作时, historyk大小并不会改变(historylength不变),仅仅移动当前指针的位置
 若当前指针在historyk的中间位置(非栈项),此时执行pushState会在指针当前的位置添加此条目,并成为新的栈项

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Documenttitle>
   <style>
        #root{
            border:1px solid red;
           height:20px;
   style>
ead>
<body>
   <div id="root">div>
   <script>
        war historythij - mandom historyy
       mindom.onpushstate - (event) -> (
           nomeda.loggovest.sype,evest.detail.state);
root.incerTTML - mindom.location.pathname;
        mindom.addDventlintener('popotate', (ovent) -> (
           manada.loggevent.type,event.statej,
            root.innerWTML - mindom.location.pattnamey
       Charatan (historym)) (
            ims oldFundState - biotory.pundStates
            interview - translate (state, title, pathone) |

interview - oldPaniState.apply(blocory, argumenta);

if (kypenf windom.orpunestate --- 'function') |
                    mindom.onpunnetate them CustomWeest ('punnetate', Idetail: [pathname, state]]]];
                 perhaps recolly
        13 (h letonyth (3 a
        imi olemiscorytengin - nistorymj.lenginy
        1 <- ()) JuneauTien
           historyShip-pussState(| page: 1 ], | title: 'paget' ], '/paget');
            {\tt nonseds.log(bistoryth).length-oldMistorytength)}_{\mathcal{S}}
        [, 1000],
        1 <- Oliventifica
            historyEmplementEace(| pages 2 ), | title: 'pages' |, '/pages');
            conside.ingmictorymj.length-oldMictorytengthja
        (, 2000),
        1 C- 1112constTino
            historyEmplementCate(| pages S ), | title: 'pages' ), '/pages');
            conside. Ingthictoryth), length-oldflictorytength),
        (, 2000),
        continuous (c) -> (
            ntecocythic backgty
            estTimeout FIT-bookede, los mistoryThis, length-oldMistorytengthi, 1991,
        1. 400001.
           historyMal-puntMate(| pagerd |, | title: 'paget' |, '/paget');
manada.log(historyMa).length-nidMatoryMength);
            historyth) - go (13 a
            manada.log(historyth).length-oldfistorytength).
```

2.基本路由#

2.1 安装 <u>#</u>

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

2.2 src\index.js

src\index.js

2.3 Home.js

src\components\Home.js

2.4 Home.js

src\components\User.is

2.4 Profile.js

src\components\Profile.js

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)

```
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(
,document.getElementById('root'));
```

3.2 RouterContext.js

 ${\it src} \ | \ {\it react-router} \ | \ {\it RouterContext.js}$

```
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();
         let value = {
              location: this. state. location.
             history: this.props.history
              <RouterContext.Provider value={value}>
{this.props.children}
              RouterContext.Provider>
    3
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;
    }
} cervort 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

3.7 HashRouter.js

src\react-router-dom\HashRouter.js

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.state;
    pathname.state;
    pathname.state;
    pathname.state;
    pathname.state;
    pathname.state;
                   pathname = pathname.pathname;
                   state=nextState;
            globalHistory.pushState(state,null,pathname);
let location = {state,pathname};
notify({action,location});
      const history = {
   action: 'POP',
            go,
goBack,
            goForward, push,
             listen,
             \texttt{location:} \{\texttt{pathname:} \textbf{window.} \texttt{location.pathname,} \textbf{state:} \textbf{window.} \texttt{location.} \textbf{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
             state = nextState;
        window.location.hash = pathname;
    window.addEventListener('hashchange', hashChangeHandler);
    function goBack() {
    function goForward() {
        go(1);
    const history = {
   action:'POP',
        go,
        goForward,
         listen.
         location:{},
        location:{pathname:'/',state:undefined}
    if (window.location.hash) {
        action = 'PUSH';
        hashChangeHandler();
        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

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 (
                      {this.props.children}
export default HashRouter;
```

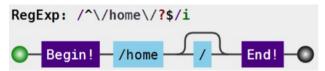
4. path-to-regexp

- regulex (https://jex.im/regulex)
- path-to-regexp (https://www.npmjs.com/package/path-to-regexp)

 - sensitive 是否大小写敏感 (對认值: 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)

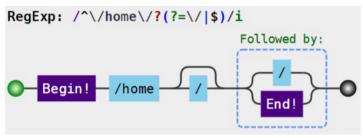
4.1 /home结束

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



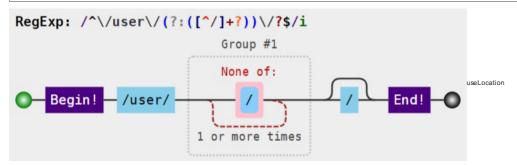
4.2 /home非结束 <u>#</u>

```
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});
 onsole.log(regExp,params);
```



4.4 正则匹配

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(?

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>

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

```
nsole.log('lab'.match(/1[a-z]([b-c])/));
     console.log('lab'.match(/1[a-z](?:[a-z])/));
          onsole.log(/(?\d{2})-(?\d{2})/.exec('11-22'));
  console.log('11-22'.replace(/(?\d{2})-(?\d{2}))/,"$-{{content}}quot;));
  表达式 含义 (?=pattem) 正向肯定查找(前瞻),后面必须跟着什么 (?!pattem) 正向否定查找(前瞻),后面不能跟着什么 ``(?<=pattem)< code> 反向肯定条件查找(后顾),不捕获 <code>(?)
                         em)</code> 反向否定条件查找(后顾)<code class="lang-js"
  < span \ class="hljs-built_in">console</span>.log(< span \ class="hljs-string">'1a'</span>.match(< span \ class="hljs-regexp">/hd(?=[a-z])(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])[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> array = [<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="hijs-literal">undefined</span>;
<span class="hijs-built_in">console</span>.log(array);
</span></=[a-z])\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]>
/=[a-z]\ld[a-z]></span></code>
/=[a-z]\ld[a-z]>
/=[a-z]\ld[a-z]\ld[a-z]>
/=[a-z]\ld[a-z]\ld[a-z]>
/=[a-z]\ld[a-z]\ld[a-z]\ld[a-z]>
/=[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-z]\ld[a-
   </a></br>
screeat-routenmatchPath.js>pre><code class="lang-js"><span class="hljs-keyword">import</span> pathToRegexp <span class="hljs-keyword">from</span> <span class="hljs-string">path-yeyword">from</span> <span class="hljs-string">path-yeyword">from</span> <span class="hljs-string">path-yeyword">from</span> <span class="hljs-string">path-yeyword">from</span> <span class="hljs-string">from</span> <span 
 to-regexp'</pan>;
<span class="hljs-function"><span class=
 <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 class="hljs-title">matchPath</span>(<span class="hljs-params">pathname.options = {}</span>(<span class="hljs-params">pathname.options = {}</span>(<span class="hljs-params")</pre>
   <span class="hijs-keyword">let</span> {path=<span class="hijs-keyword">let</span> {path=<span class="hijs-kiring">"</span>,exact=<span class="hijs-kiring">let</span>,strict=<span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span>,strict=<span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">span class="hijs-kiring">let</span> {path=<span class="hijs-kiring">let</span class="hijs-kiring">let</s
literal">false</span>}=options:
   <span class="hljs-keyword">let</span> {keys,regexp} = compilePath(path,{
   <span class="hlis-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>
  <soan class="hlis-keyword">const</soan> [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>{
                                     memo[key.name] = values[index];
<span class="hljs-keyword">return</span> memo;
                    },{})
  <span class="hljs-keyword">export</span> <span class="hljs-keyword">efault</span> matchPath;
</code><h3 id="t375.2 Router.js">5.2 Router.js">5.2 Router.js<5.2 Router.js<5.2 Router.js">5.4 Router.js<5.2 Router.js<5.2 Router.js<5.2 Router.js<6.2 Router.js<6.2
 import RouterContext from './RouterContext';
 class Router extends React.Component(
  class Notice exterior reaction reaction printing (span) (s
   <span class="hljs-addition">+ }</spa</pre>
 constructor(props)(
  super(props);
 this.state = {
 location:props.history.location
  ,
//当路径发生的变化的时候执行回调
 this.unlisten = props.history.listen((location)=>{
 this.setState({location});
 }):
 componentWillUnmount()
  this.unlisten&&this.unlisten();
 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)
   <routercontext.provider value="{value}">
  {this.props.children}
   </routercontext.provider
  </code><h3 id="t385.3 Route.js">5.3 Route.js a href="#t385.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js"<#t395.3 Route.js">#t395.3 Route.js"<#t395.3 Route.js"</td>
#t395.3 Route.js"<#t395.3 Route.js</td>
#t395.3 Route.js
#t395.3 Route.js
#t395.3 Route.js
#t395.3 Route.js
#t395.3 Route.js
#t395.3 Route.
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>
                                                   component {...routeprops}>
  <routercontext.provider value="{routeProps}">
  </routercontext.provider>
 export default Route;
  </ri>
</ri>
//routecomponent/</code><h3 id="t395.4 react-router\index.js">5.4 react-router\index.js<a href="#t395.4 react-router\index.js">#t</a>//routecomponent/
diff">export {default as Route} from './Route';
export (default as Router) from './Router';
export (default as ___RouterContext) from './RouterContext';
 routen/Switch.js>rov=code class="lang-js"><span class="hljs-keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span> RouterContext <span class="hljs-keyword">import</span> RouterContext <span class="hljs-keyword">from</span> <span clas
 <pan class="hljs-keyword">import</span> matchPath <span class="hljs-keyword">from</span> <span class="hljs-keing">'./matchPath'</span>;
<pan class="hljs-keyword">extends</span> <span class="hljs-keyword">extends</span> <span
title">Component</span></span>\{
 <span class="hlis-keyword">static</span> contextType = RouterContext
render() {
  <span class="hljs-keyword">const</span> { location } = <span class="hljs-keyword">this</span>.context;
<span class="hljs-keyword">let</span> element, match;
React.Children.forEach(<span class="hljs-keyword">this</span>.props.children, child => {
  <span class="hljs-keyword">if</span> (!match && React.isValidElement(child)) {
element = child;
match = matchPath(location.pathname, child.props);
  .//
<pan class="hljs-keyword">retum</span> match? React.cloneElement(element, {<span class="hljs-attr">computedMatch</span>: match }) : <span class="hljs-literal">null</span>
 , <span class="hljs-keyword">export</span> <span class="hljs-keyword">export</span> <span class="hljs-keyword">export</span> <span class="hljs-keyword">export</span> <span class="lang-diff">switch; </code></span class="hljs-keyword">export</span> <span class="lang-diff">import React from 'react' </span> <span class="lang-diff">import React from 'react' </span 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}>
  <routercontext.provider value="{routeProps}">
 {element}
  </routercontext.provider>
 /routecomponent></code><h3 id="t436.3 react-router\index.js">6.3 react-router\index.js <a href="#t436.3 react-router\index.js">#</a></h3>src\react-router\index.js
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 (uerant as maturi-any nom: ./maturi-ann, cs-span) (sessimity-bayotion) ./maturi-ann, cs-span) (sessimity-bayotion) ./maturi-ann, cs-span) (sessimity-bayotion) ./maturi-ann, cs-span) ./span) .
 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(
  <span class="hlis-addition">+ <switch></switch></span>
<route path="/" component="{Home}" exact>
<route path="/user" component="{User}">
  <ruet path="/profile" component="{Profile}/">
  <span class="hljs-addition">+ </span>
  </route></route></route>
 .document.aetElementBvld('root'));

/code>
/pe>
1d="1457. Redirect">7. Redirect <a href="#4457. Redirect">#</a>
/p>
/p2

1d="1457. Redirect.js">a href="#467.1 Redirect.js">#
/p3
/p3
/p3

routen/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.jsrouten/Redirect.js
<span class="hijs-keyword">import</span> RouterContext <span class="hijs-keyword">from</span> <span class="hijs-kering">'./RouterContext'</span>;<span class="hijs-keyword">import</span> Lifecycle <span class="hijs-keyword">from</span> <span class="hijs-keyword">import</span>;
  <span class="hljs-function"><span class="hljs-key</pre>
                                                                                                                                                     rd">function</span> <span class="hljs-title">Redirect</sp
                                                                                                                                                                                                                                                                                                                       an>(<span class="hljs-params">{to}</span>)</span>{
  <span class="hljs-keyword">return</span> (
  <span class="xml"><span class="hljs-tag"><<span class="hljs-name">RouterContext.Consumer</span>></span>
 contextValue=>{
const {history}= contextValue;
 return (
  <span class="hljs-tag"><<span class="hljs-name">Lifecycle</span>
  <span class="hljs-attr">onMount</span>=<span class="hljs-string">{()</span>=></span>history.push(to)}
  ,
<span class="hljs-tag">RouterContext.Consumer</span>></span>
```

export default Redirect;

```
</code><h3 id="t477.2 Lifecycle.js">7.2 Lifecycle.js <a href="#t477.2 Lifecycle.js">#</a></h3>srctreact-router/Lifecycle.js<code class="lang-js"><span class="hljs</p>
  keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-string">'react'</span>;
   <span class="hljs-class"><span class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-key
  title">Component</span></span>{
 componentDidMount(){
   <span class="hljs-keyword">if</span>(<span class="hljs-keyword">this</span>.props.onMount)
   <span class="hljs-keyword">this</span>.props.onMount(<span class="hljs-keyword">this</span>);
 componentWillUnmount()
  <span class="hljs-keyword">if</span>(<span class="hljs-keyword">if</span>.props.onUnmount)
<span class="hljs-keyword">this</span>.props.onUnmount(<span class="hljs-keyword">this</span>;props.onUnmount(<span class="hljs-keyword")</p>
 render(){
   <span class="hljs-keyword">return</span> <span class="hljs-literal">null</span>;
   .
<span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> Lifecycle;
 </code><h3 id="t487.3 react-routerlindex.js">7.3 react-routerlindex.js = 7.3 react-routerlindex
 export {default as Router} from './Router';
export {default as __RouterContext} from './RouterContext';
 export {default as matchPath} from './matchPath'; export {default as Switch} from './Switch';
  <span class="hljs-addition">+export {default as Redirect} from './Redirect';</span>
</code></dia="t497.4 src\index.js">7.4 src\index.js">7.4 src\index.js">1.4 src\
 import ReactDOM from 'react-dom';
   <span class="hljs-addition">+import {HashRouter as Router,Route,Switch,Redirect} from './react-router-dom';</span>
import Home from './components/Home' import User from './components/User';
 import Profile from './components/Profile';
 ReactDOM.render(
   <router>
  <switch>
  <route path="/" component="{Home}" exact>
   <route path="/user" component="{User}">
  <route path="/profile" component="{Profile}">
<span class="hljs-addition">+ <redirect to="/"></redirect></span>
  </raute></raute></raute></switch>
   </router>
  ...document.getElementByld("root"));
</code>srctreact-router-dom\Link.js<code class="lang-js"><span class="hljs-keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-keyword">from</span c
  string">'react'</span>;
   <span class="hijs-keyword">import</span> {_RouterContext <span class="hijs-keyword">as</span> RouterContext <span class="hijs-keyword">from</span> <span class="hijs-keyword">from</span class="hijs-keyword">from</span class="hijs-keyword">from</span 
 <span class="hljs-keyword">export</span> <span class="hljs-keyword">function</span> <span class="hljs-keyword">function</span class="hljs-keyword">function</span class="hljs-keyword">function</span class="hljs-keyword">function</span c
  <span class="hljs-keyword">return</span> (
   <span class="xml"><span class="hljs-tag"><<span class="hljs-name">RouterContext.Consumer</span>></span</pre>
 contextValue=>{
 return (
   <span class="hljs-tag"><<span class="hljs-name">a</span>
 {<span class="hljs-attr">...props</span>}
<span class="hljs-attr">onClick</span>=<span class="hljs-string">{(event)</span>=></span>}
  event.preventDefault():
 contextValue.history.push(props.to);
   ,,
>{props.children}<span class="hljs-tag">a</span>></span>
   <span class="hljs-tag">RouterContext.Consumer</span>>
   </code><h3 id="t528.2 react-router-dom\index.is">8.2 react-router-dom\index.is<<a href="#t528.2 react-router-dom\index.is">#</a></h3>src\react-router-dom\index.is<code class="lang."
   diff">export * from '../react-router';
  export (default as HashRouter) from './HashRouter'
   export {default as BrowserRouter} from './BrowserRouter';
   <span class="hlis-addition">+export {default as Link} from './Link':</span>
   </code></h3 id="t538.3 src\index.js">8.3 src\index.js > 8.3 src\index.js > 4 href="#t538.3 src\index.js">#</a></h3>src\index.js/p>code class="lang-diff">import React from 'react';
 import ReactDOM from 'react-dom'
    <span class="hljs-addition">+import {HashRouter as Router,Route,Switch,Redirect,Link} from './react-router-dom';</span>
 import Home from './components/Home'; import User from './components/User';
 import Profile from './components/Profile'
  ReactDOM.render(
   <router>
   <span class="hljs-addition">+ </span>
   <span class="hljs-addition">+ kto="/">首页</span>
  <span class="hljs-addition">+ <link to="/user">用户管理<span class="hljs-addition">+ <link to="/profile">个人中心
   <span class="hljs-addition">+ </span>
  <switch>
   <route path="/" component="{Home}" exact>
   <route path="/user" component="{User}">
   <route path="/profile" component="{Profile}/">
  <redirect to="/">
  </router>
  ,document.getElementByld("root"));
</code><h2 id="t549.嵌套路由">8.嵌套路由 <a href="#t549.嵌套路由">#</a></h2><h3 id="t559.1 utils.js">9.1 utils.js">9.1 utils.js">9.1 utils.js">#</a></h3>src\utils.js">#</a></h3>src\utils.js
 js"><span class="hljs-keyword">export</span> <span class="hljs-keyword">const</span> UserAPI = {
 list(){
  *span class="hljs-keyword">let</span> usersStr = localStorage.getItem(<span class="hljs-ktring">users</span>);
*span class="hljs-keyword">let</span> users= usersStr ? <span class="hljs-built_in">JSON</span>.parse(usersStr): [];
   <span class="hljs-keyword">return</span> users;
   add(user){
   <span class="hljs-keyword">let</span> users = UserAPI.list();
   users.push(user);
 localStorage.setItem(<span class="hlis-string">'users'</span>. <span class="hlis-built in">JSON</span>.stringify(users));
  find(id){
   <span class="hljs-keyword">let</span> users = UserAPI.list();
   <span class="hljs-keyword">retum</span> users.find(<span class="hljs-function">(<span class="hljs-params">user</span>) =></span> user.id === id);
```

```
.
</code><h3 id="t569.2 User,is">9.2 User,is <a href="#t569.2 User,is">#t569.2 User,is">#t569.2 User,is">#t569.2 User,is">#t569.2 User,is <a href="#t569.2 User,is">#t569.2 User,is <a href="#t5
 import { Link, Route } from '../react-router-dom'; 
<span class="hljs-addition">+import UserAdd from './UserAdd';</span>
  <span class="hijs-addition">+import User/Detail from './User/Detail;<span class="hijs-addition">+import User/List from './User/List';
   export default function User() {
 return (
    <span class="hljs-addition">+ </span
   <span class="hlis-addition">+ </span>
  <span class="hljs-addition">+ </span>
   <span class="hljs-addition">+ </span>
    <span class="hljs-addition">+ <route path="/user/add" component="{UserAdd}"></route></span>
   <span class="hijs-addition">+ <route path="/user/list" component="{UserList}"></route></span>
   <span class="hljs-addition">+ <route path="/user/detail/:id" component="{UserDetail}"></route></span>
   <span class="hljs-addition">+ </span>
   <span class="hljs-addition">+ </span>
   , </ri>

<t
 React, {Component} < span class="hljs-keyword">from</span> < span class="hljs-string">'react'</span>; < span class="hljs-keyword">import</span> {UserAPI} < span class="hljs-keyword">from</span> < span class="hljs-keyword">import</span> {UserAPI} < span class="hljs-keyword">from</span> 
<span class="hijs-keyword">exportspan class="hijs-ke
  usemameRef
   <span class="hljs-keyword">constructor</span>(props) {
  <span class="hljs-keyword">super</span>(props);
<span class="hljs-keyword">this</span>.usemameRef = React.createRef();
  handleSubmit = <span class="hljs-function">(<span class="hljs-params">event</span>) =></span> {
  event.preventDefault();
   <span class="hljs-keyword">let</span> username = <span class="hljs-keyword">this</span>.usernameRef.current.value
 UserAPI.add(( 'span class="hljs-sttr">id</span>: <span class="hljs-built_in">Date</span>.now() + <span class="hljs-string">"</span>, username }); <span class="hljs-string">"</span>, username });
  render() {
   <span class="hljs-keyword">return</span> (
   {this.usernameRef}</span> /></span>
   - <span class="hljs-tag"><<span class="hljs-hame">button</span> <span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag">></span>提交<span class="hljs-tag">button</span> <span class="hljs-tag">></span> と/span>提交<span class="hljs-tag">></span> と/span> 提交<span class="hljs-tag">></span> と/span> と/sp
   <span class="hljs-tag">form</span>>
   . </code><h3 id="t589.4 UserList.js">9.4 UserList.js">9.4 UserList.js <a href="#t589.4 UserList.js">=/#t589.4 UserList.js">=/#t589.
 React, {Component } <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span> {Link} <span class="hljs-keyword">import</span> {Link} <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span> {Link} <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span> (span>);
  class="hljs-keyword">extends</span> <span class="hljs-title">Component</span> </span>{
  state = { <span class="hljs-attr">users</span>: [] }
 componentDidMount() {
   <span class="hljs-keyword">let</span> users = UserAPI.list();
   <span class="hljs-keyword">this</span>.setState({ users })
  render() {
   <span class="hljs-keyword">return</span> (
   <span class="xml"><span class="hljs-tag"><<span class="hljs-name">ul</span> ></span>
  this state users man((user index) => (
   <span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><(index)</span>></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></tp>
  <pan class="hljs-tag"><<pan class="hljs-tag"><<pan class="hljs-tatr">span class="hljs-
   </span>{user.usemame}<span class="hlis-tag">Link</span>></span>
   <span class="hljs-tag">li</span>>
   <soan class="hlis-tag">ul</soan>>
   , </code><h3 id="t599.5 UserDetail.is">9.5 UserDetail.is">9.5 UserDetail.is<a href="#t599.5 UserDetail.is">#t599.5 UserDetail.is"<#t599.5 UserDetail.is">#t599.5 UserDetail.is<#t599.5 UserDetail.is<#t
   keyword">import</span> React, { Component } <span class="hljs-keyword">from</span> <span class="hljs-string">'react'</span>
  <span class="hljs-keyword">import</span> {UserAPI} <span class="hljs-keyword">from</span> <span class="hljs-kring">'../utils'</span>;
    span class="hljs-keyword">exponts/span> <span class="hljs-keyword">defaults/span> <span class="hljs-keyword">class="hljs-keyword">class=/span class=/span> <span class="hljs-keyword">cspan class=/span> <span class="hljs-keyword">cspan class=/span> <span class="hljs-keyword">cspan class=/span> <span class="hljs-keyword">cspan class=/span> <span class=/spa
 class="hlis-keyword">extends</span> <span class="hlis-title">Component</span> </span>{
   <span class="hlis-attr">user</span>: {}
  componentDidMount() {
   <span class="hljs-keyword">let</span> user = <span class="hljs-keyword">this</span>.props.location.state;
   <span class="hljs-keyword">if</span> (!user) {
   <span class="hljs-keyword">let</span> id = <span class="hljs-keyword">this</span>.props.match.para
 user = UserAPI.find(id):
   ,
<span class="hljs-keyword">if</span> (user) <span class="hljs-keyword">this</span>.setState({ user });
  render() {
   <span class="hljs-keyword">let</span> user = <span class="hljs-keyword">this</span>.state.user;
   <span class="hlis-keyword">return</span> (
   <span class="xml"><span class="hljs-tag"><<span class="hljs-name">div</span>
  {user.id}:{user.usemame}
    <span class="hljs-tag">div</span>></span>
   .
/-(code><h2 id="t6010.受保护路由">10.受保护路由 <a href="#t6010.受保护路由">#c/a></h2><h3 id="t6110.1 sro\index.js">10.1 sro\index.js">10.1 sro\index.js <a href="#t6110.1 sro\index.js">#c/a></h3>
   src\index.js<code class="lang-diff">import React from 'react';
 import ReactDOM from 'react-dom'
   <span class="hljs-addition">+import {HashRouter as Router,Route,Switch,Redirect,Link} from './react-router-dom';</span>
```

```
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>
  <111>
  i>i>io
 = "/user">用户管理+ 小
  <switch>
  <route path="/" component="{Home}" exact>
  <route path="/user" component="{User}">
  <span class="hijs-addition">+ rotected path="/profile" component="{Profile}">/span class="hijs-addition">+ <route path="/login" component="{Login}/">/rote//span class="hijs-addition">+ <route path="/login" component="{Login}/">/rote//span class="hijs-addition">+ /route//span class="hijs-addition">+ /route//<pre
   <redirect to="/">
   </redirect></route></route></switch>
  </router>
   ,document.getElementById('root'));
 </code><h3 id="f6210.2 Route.js">10.2 Route.js">10.2 Route.js">10.2 Route.js">4 ref="#f6210.2 Route.js">#</a></h3>srcireact-router/Route.js// Route.js">4 ref="#f6210.2 Route.js
   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="hijs-addition">+ element = <routecomponent {...routeprops}></routecomponent></span>
<span class="hijs-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><h3 id="t6310.3 Login.js">10.3 Login.js">10.3 Login.js<herriv</p></code>
  <span class="hljs-keyword">from </span> < span class="hljs-string">'react'</span>;
<span class="hljs-class"><span class="hljs-class"><span class="hljs-keyword">extends</span> < span class="hljs-tlitle">React</span>.<span class="hljs-tlitle">React<
 title">Component</span></span>{
 login = <span class="hljs-function"><span class="hljs-params">()</span>=></span>{
localStorage.setItem(<span class="hljs-string">"login'</span>,<span class="hljs-string">"true'</span>); <span class="hljs-keyword">let</span> to=<span class="hljs-string">/"</span>;
 class="hljs-keyword">this</span>.props.history.push(to);
  <span class="hljs-keyword">return</span> (
    -(span class="xni"><span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><(span class="hijs-tag"><<span class="hijs-tag"><(span cla
 tag">button</span>></span>
   <span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> Login;
   </code><h3 id="t6410.4 Protected.js">10.4 Protected.js <a href="#t6410.4 Protected.js">+8.4 href="#t64
 keyword">import</span> React <span class="hlis-keyword">from</span> <span class="hlis-string">react </span>:
   <span class="hljs-keyword">import</span> {Route,Redirect} <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span>;"../react-router-dom'</span>;
   <soan class="hlis-keyword">const</soan> Protected = <soan class="hlis-function">(<soan class="hlis-params">props</soan>)=></soan>
   <span class="hljs-keyword">let</span> {<span class="hljs-attr">component</span>:RouteComponent,path} = props;
  <span class="hljs-keyword">return</span> (
   <qpan class="xml"><span class="hijs-tag"><<span c
 class="hlis-string">{</span>
  (<span class="hljs-attr">routeProps</span>)=></span>(
 localStorage.getItem('login')?<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"></span class="hljs-tag"><
   <span class="hljs-tag"><<span class="hljs-tag"></span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"></span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag"><<span class="hljs-tag">
 attr">state:</span>{<span class="hljs-attr">from:path</span>}}}/></span>
  \/></soan>
   -
<span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> Protected;
   </code><h3 id="t6611.1 public\index.html">11.1 public\index.html">11.1 public\index.html"></code><h3 id="t6611.1 public\index.html"/p><code class="lanq-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>
 <body>
 </body>
  </html>
 </code><h3 id="t6711.2 src\index.js">11.2 src\index.js<11.2 src\index.js<a href="#t6711.2 src\index.js">#</a></h3>src\index.js/po><code class="lang-diff">import React from 'react';
import ReactDOM from 'react-dom';
Import Reactive into the activation "state of the state o
import User from './components/User';
import Profile from './components/Profile';
import Protected from './components/Protected':
import Login from './components/Login';
ReactDOM.render(
  <router>
 <span class="hljs-addition">+ <navlink classname="strong" style="{(textDecoration:" 'line-through'}} activestyle="{(color:'red'})" to="/" exact>Home</navlink></span:
 <span class="hlis-addition">+ <navlink activestyle="{{color:'red"}}" to="/user">User</navlink></span>
  <span class="hljs-addition">+ <navlink activestyle="{{color:'red'}}" to="/profile">Profile</navlink></span>
 <route path="/" component="{Home}" exact>
  <route path="/user" component="{User}">
  <route path="/login" component="{Login}/">
 <redirect to="/">
  </redirect></route></protected></route></route></switch>
 </router>
  ,document.getElementByld('root'));
 string">'react'</span>;
 <span class="hljs-keyword">import</span> {Route} <span class="hljs-keyword">from</span> <span class="hljs-keyword">.../react-router'</span>;
<span class="hljs-keyword">import</span> matchPath <span class="hljs-keyword">from</span> <span class="hljs-keyword">.../react-router/matchPath'</span>;
 <span class="hljs-keyword">import</span> Link <span class="hljs-keyword">from</span> <span class="hljs-string">'./Link</span>;
<span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span class="hljs-function"><span>(span class="hljs-function")
 <span class="hljs-keyword">const</span> {
  <span class="hljs-attr">to</span>:path,
 <span class="hljs-attr">className</span>:classNameProp=<span class="hljs-string">"</span>
  <span class="hljs-attr">style</span>:styleProp={},
activeClassName=<span class="hlis-string">'active'</span>.
  activeStyle={},
children.
 exact
}= props
  <span class="hljs-keyword">return</span> (
 <span class="xml"><span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tatr">path</span>=<span class="hijs-tring">(path)</span> <span class="hijs-tring">(path)</span class="hijs-tring">(path)</span class="hijs-tr
  class="hljs-string">{exact}</span>></span>
let className = match?ioinClassNames(classNameProp.activeClassName):classNameProp
let style =match?{...styleProp,...activeStyle}:styleProp;
let linkProps = {
 className,
style.
to:path
children
 .
returm <span class="hljs-tag"><<span class="hljs-name">Link</span > {<span class="hljs-attr">...linkProps</span>}/></span>
  <span class="hljs-tag">Route</span>></span>
  <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">NavLink</span>(<span class="hljs-params">props</span>)</span>[
<span class="hljs-keyword">let</span> context = React.useContext(RouterContext);
<span class="hljs-keyword">let</span> {<span class="hljs-attr">location</span>:{path
 <span class="hljs-keyword">const</span> {
  <span class="hljs-attr">to</span>:path,
 $$ \exp \alpha class="hijs-attr">className</span>:classNameProp=<span class="hijs-string">"</span>, <span class="hijs-attr">style</span>:styleProp={},
activeClassName=<span class="hljs-string">'active'</span>,
  activeStyle={},
children.
exact
}= props
    <span class="hljs-keyword">let</span> isActive = matchPath(pathname, {path,exact});
    <span class="hljs-keyword">let</span> className = isActive?joinClassNames(classNameProp,activeClassName):classNameProp;<span class="hljs-keyword">let</span> style = isActive?{...styleProp,...activeStyle}:styleProp;
    <span class="hljs-keyword">let</span> linkProps = {
             className,
             style,
<span class="hljs-attr">to</span>:path,
              children
    <span class="hljs-keyword">return</span> <span class="xml"><span class="hljs-tag"><<span class="hljs-name">Link</span> {<span class="hljs-</pre>
  attr">...linkProps</span>}/></span></span>;
 <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-function"><span clas
  <span class="hljs-keyword">retum</span> class=mnljs-keyword">retum</span> class=mnljs-keyword">retum</span> class=mnljs-keypan>class=mnljs-keypan> class=mnljs-keypan> class=mnljs-ke
  ,
<span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> NavLink;
</code></h3/id="t6911.4 react-router-dom\index.js">11.4 react-router-dom\index.js">11.4 react-router-dom\index.js">4
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><h2 id="t7012.withRouter">12.withRouter">12.withRouter<h12.withRouter</p>
*/a></h2><h3 id="t7112.1 src\index.js">12.1 src\index.js
*src\index.js">12.1 src\index.js
*src\index.js">p>*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="hlis-addition">+ <navheader title="欢迎光临"></navheader></span>
  <navlink classname="strong" style="{{textDecoration:" 'line-through'}} activestyle="{{color:'red'}}" to="/" exact>Home</navlink>
 <ii><navlink activestyle="{(color:red')}" to="/user">User</navlink></ii>
<navlink activestyle="{(color:red')}" to="/profile">Profile</navlink>

  <switch>
  <rue><route path="/" component="{Home}" exact></ri>
   <route path="/user" component="{User}">
   component="{Profile}/">
   <route path="/login" component="{Login}/">
  <redirect to="/">
   </redirect></route></protected></route></switch>
   </>
  </router>
   ,document.getElementByld('root'));
   </code>src/components/NavHeader.js<code class="lang-js"><span class="hljs-keyword">import</span> React <span class="hljs-keyword">from</span> <span class="hljs-keyword">from</span class="hljs-keyword">from</s
  string">'react'</span>
   <span class="hljs-keyword">import</span> {withRouter} <span class="hljs-keyword">from</span> <span class="hljs-keyword">../react-router-dom'</span>;
<span class="hljs-class"><span class="hljs-keyword">cxpan class="hljs-keyword">extends</span> <span class="hljs-keyword">extends</span class="hljs-keyword">extends</span class="hljs-keyword">extends</span class="hljs-keyword">extends</span class="hljs-keyword">extends</span cl
 render(){
   <span class="hljs-keyword">return</span> (
   <span class="xml"><span class="hijs-tag"><<span class="hijs-tag"><</p>
  {this.props.title}<span class="hljs-tag">div</span>></span>
   <span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> withRouter(NavHeader);
 </code><h3 id="t7312.3 withRouter.js">12.3 withRouter.js">12.3 withRouter.js">12.3 withRouter.js">4 c/code>*/p><code class="lang-js"><span class="hljs-keyword">import/span> React <span class="hljs-keyword">import/span> RouterContext <span class="hljs-keyword">import/span class="hljs-keyword">import/span> RouterContext <span class="hljs-keyword">import/span class="
   <span class="hljs-function"><span class="hljs-keyword">function</span> (span class="hljs-function"><span clas
   <span class="hljs-keyword">returm</span> <span class="hljs-function"><span class="hljs-params">props</span> =></span> {
   <span class="hljs-keyword">return</span> (
   <span class="xml"><span class="hljs-tag"><<span class="hljs-name">RouterContext.Consumer</span>></span>
  contextValue =>{
  retum <span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"></span class="hijs-tag"><
   -
<span class="hljs-tag">RouterContext.Consumer</span>></span>
   <span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> withRouter;
   </code></3 id="t7412.4 react-routerlindex.js">12.4 react-routerlindex.js <a href="#t7412.4 react-routerlindex.js">+</a>>>
  {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'
   <span class="hljs-addition">+export {default as withRouter} from './withRouter';</span>
  </code><h2 id="t7513.Prompt">13.Prompt"
4 href="#t7513.Prompt">#</a></h2><h3 id="t7613.1 srclindex.js">13.1 srclindex.js
4 href="#t7613.1 srclindex.js">#</a>
4 href="#t7613.1 srclindex.js">#</a>
   <code class="lang-diff">import React from 'react';
 import ReactDOM from 'react-dom':
   <span class="hljs-addition">+import {BrowserRouter 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';
 import NavHeader from './components/NavHeader';
 ReactDOM.render(
   <span class="hljs-addition">+ <router getuserconfirmation="{()=">window.confirm}></router></span>
  <|i><navlink classname="strong" style="{{textDecoration:" 'line-through'}} active style="{{color:'red'}}" to="/" exact>Home</navlink>
  <navlink activestyle="{{color:'red'}}" to="/user">User</navlink>
   <navlink activestyle="{{color:'red'}}" to="/profile">Profile</navlink>
  <switch>
  <route path="/" component="{Home}" exact>
   <route path="/user" component="{User}">
  cprotected path="/profile" component="{Profile}/">
 <route path="/login" component="{Login}/">
   </redirect></route></route></route></switch>
   ,document.getElementByld("root"));
</navheader></code></sid="t7713.2 UserAdd.js">13.2 UserAdd.js">13.2 UserAdd.js <a href="#t7713.2 UserAdd.js">#t7713.2 UserAdd.js"<#t7713.2 UserAdd.js">#t7713.2 UserAdd.js">#t7713.2 UserAdd.js"<#t7713.2 UserAdd.js">#t7713.2 UserAdd.js"<#t7713.2 UserAdd.js"<#t7713.2 UserAdd.js">#t7713.2 UserAdd.js"<#t7713.2 UserAdd.js"<#t7713
 Component } from 'react'
 import {UserAPI} from '../utils';
import {Prompt} from '../react-router-dom';
 export default class UserAdd extends Component {
   usemameRef
   <span class="hljs-addition">+ state = {isBlocking:false}</span>
  constructor(props) {
 super(props);
```

```
this.usernameRef = React.createRef();
handleSubmit = (event) => {
 event.preventDefault();
  <soan class="hlis-addition">+ this.setState({</span>
 <span class="hljs-addition">+ isBlocking:false</span
<span class="hljs-addition">+ },()=>{</span>
let usemame = this.usemameRef.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">+ rompt< span>
<span class="hljs-addition">+ when={this.state.isBlocking}</span>
 <span class="hljs-addition">+ }</span>
  <span class="hljs-addition">+ /></span>
 <span class="hijs-addition">+ <input type="text" ref="(this.usemameRef)" onchange="((event)" => {</span>
<span class="hijs-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><h3 id="t7813.3 Prompt.js">13.3 Prompt.js<a href="#t7813.3 Prompt.js">#s/a>rc/react-router/Prompt.js
React <span class="hljs-keyword">from</span> <span class="hljs-string">'react'</span>;
 <span class="hljs-keyword">import</span> RouterContext <span class="hljs-keyword">from</span> <span class="hljs-keyword">import</span> LifeCycle <span class="hljs-keyword">import</span> LifeCycle <span class="hljs-keyword">import</span> LifeCycle <span class="hljs-keyword">import</span>;
 <span class="hljs-function"><span class="hljs-functio
  <span class="hljs-keyword">let</span> value = React.useContext(RouterContext);
React.useEffect(<span class="hljs-function"><span class="hljs-params">()</span>=></span>{
  <span class="hljs-keyword">return</span> value.history.block(props.me
}):
  <span class="hljs-keyword">return</span> <span class="hljs-literal">null</span>;
 <span class="hijs-class"><span class="hijs-keyword">class<span class="hijs-keyword">class<span class="hijs-keyword">extends</pan> <span class="hijs-keyword">expan cla
title">Component</span></span>{
 <span class="hljs-keyword">static</span> contextType = RouterContext;
componentDidMount(){
 <span class="hijs-keyword">const</span> block = <span class="hijs-keyword">this</span>.context.history.block
<span class="hijs-keyword">this</span>.release = block(<span class="hijs-keyword">this</span>.props.message)
componentWillUnmount(){
 <span class="hljs-keyword">this</span>.release()
 render(){
  <span class="hljs-keyword">return</span> <span class="hljs-literal">null</span>;
<span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-function"><span class="hljs-params">{\span class="hljs-params">\span>{\span class="hljs-params">\span>{\span class="hljs-function"><\span class="hljs-function">\span>{\span class="hljs-function"><\span class="hljs-function">\span>{\span class="hljs-function"><\span class="hljs-function">\span>{\span class="hljs-function"><\span class="hljs-function">\span>{\span class="hljs-function"><\span class="hljs-function">\span>{\span class="hljs-function">\span class=\span class=\span
  <span class="hljs-keyword">return</span> (
 <span class="xml"><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;
  <span class="hlis-tag"><<span class="hlis-name">LifeCvcle</span>
 <span class="hljs-attr">onMount</span>=<span class="hljs-string">(lifeCycleInstance</span>=></span>lifeCycleInstance.release = block(message))
onUnMount={lifeCvcleInstance=>lifeCvcleInstance.release()}
             <span class="hljs-tag">RouterContext.Consumer</span>></span>
  <span class="hljs-keyword">export</span> <span class="hljs-keyword">default</span> Prompt;
  </code><h3 id="t7913.4 react-router\index.js">13.4 react-router\index.js">13.4 react-router\index.js">4 react-router\index.js">#</a><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 './withRoute
 <span class="hlis-addition">+export {default as Prompt} from './Prompt':</span>
  </code><h3 id="t8013.5 createBrowserHistory.jg">13.5 createBrowserHistory.js</a href="#t8013.5 createBrowserHistory.jg">*/a>
diff">
 function createBrowserHistory(props){
 <span class="hljs-addition">+ let confirm =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()
```

```
,
function listen(listener){
listeners.push(listener);
  return function(){//unliste
listeners = listeners.filter(l=>l!==listener):
  function setState(newState){
Object.assign(history,newState):
 history.length = globalHistory.length
listeners.forEach(listener=>listener(history.location));
  * @param {} pathname 可能是对象,也可能是字符串
* @param {} state 这个路径的状态对象是什么,只是一个路径的描述信息,可以放任何
function push(pathname, state){
 const action = 'PUSH';//表示发了什么动作引起了路径变化 POP PUSH
 if(typeof pathname
 state=pathname.state
pathname= pathname.pathname
  <span class="hljs-addition">+ if(message){</span>
 <span class="hijs-addition">+ let showMessage = message({pathname});
<span class="hijs-addition">+ let allow = confirm(showMessage);</span>
                                                                                                                                             sage({pathname});</span>
  <span class="hljs-addition">+ if(!allow){</span>
  <span class="hljs-addition">+ return;</span>
 <span class="hljs-addition">+ }</span>
<span class="hljs-addition">+ }</span>
 globalHistory.pushState(state,null,pathname);
let location = {state,pathname};
 setState({action,location});
  ,
<span class="hljs-addition">+ function block(newMessage){</span>
 <span class="hljs-addition">+ message=newMessage;</span>
  <span class="hljs-addition">+ return ()=> message=null;</span>
  <span class="hljs-addition">+ }</span>
const history = {
action: "POP",
go,
goBack,
  goForward
listen,
location:{pathname:window.location.pathname,state:globalHistory.state},
push.
  <span class="hljs-addition">+ block</span>
  return history:
export default createBrowserHistory:
  </code><h2 id="t8115.hooks">15.hooks <a href="#t8115.hooks">#</a></h3>srclindex.jsfoode><ipre><h2 id="t815.1 srclindex.js</p>foode>
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> ReactDOM <span class="hljs-keyword">from</span> <span class="hljs-tring">ingort</span> (ass="hljs-tring">ingort</span> (ass="hljs-tring") react-dom'</span>
  <span class="hlis-keyword">import</span> {BrowserRouter <span class="hlis-keyword">as</span> Router.Route.Link
useParams
useLocation.
 useHistory,
useRouteMatch
 } <span class="hljs-keyword">from</span> <span class="hljs-string">'./react-router-dom'</span>;
<span class="hljs-function"><span class="hljs-function"><span> <span class="hljs-title">Home</span>()</span>{
  <span class="hljs-keyword">retum</span> <span class="xml"><span class="hljs-tag"><<span class="hljs-name">div</span>><ispan>if页<span class="hljs-tag">div</span><ispan>
  ,
<span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">UserDetail</span>()</span>{
  <span class="hljs-keyword">let</span> params = useParams();
 <span class="hijs-built_in">console</span>.log(<span class="hijs-string">'params'</span>,params);<span class="hijs-keyword">let</span> location = useLocation();
  <span class="hljs-built_in">console</span>.log(<span class="hljs-string">'location'</span>,locat
  <span class="hlis-keyword">let</span> history = useHistory();
 name">br</span>/></span>name:{location.state.name}<span class="hljs-tag">div</span>></span>;
 <span class="hljs-function"><span class="hljs-title">Post</span>(
  <span class="hljs-keyword">let</span> match = useRouteMatch({
  <span class="hlis-attr">path</span>:<span class="hlis-string">'/post/:id'</span>
  <span class="hljs-attr">strict</span>:<span class="hljs-literal">true</span>
 <span class="hljs-attr">sensitive</span>:<span class="hljs-literal">true</span>
  <span class="hljs-keyword">returm</span> match?span class="hmilly-span class="hijs-tag">span class="hijs-keyword">returmspan class="hijs-tag">span class="
  <span class="xml"><span class="hljs-tag"><<span class="hljs-name">div</span>></span>Not Found<span class="hljs-tag">div</span>></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></tp>
 ReactDOM.render(
  <span class="xml"><span class="hljs-tag"><<span class="hljs-name">Router</span>></span>
 <span class="hljs-tag"><<span class="hljs-name">div</span>></span>
<span class="hljs-tag"><<span class="hljs-name">ul</span>></span></span></span></pa>
  <span class="hlis-tag"><<span class="hlis-tag"><</p>
  </span>首页<span class="hljs-tag">Link</span>></span><span class="hljs-tag">li</span>>
  <gpan class="hijs-tag"><<span class="hijs-tag"></span class="hijs-tag"><<span class="hijs-tag"><<
  attr">name:</span>'珠峰架构'}}}></span>用户1详情<span class="hljs-tag">Link</span>><span class="hljs-tag">li</span>>
   -span class="hljs-tag"><<span 
  </span>></span>贴子<span class="hljs-tag">Link</span>><span class="hljs-tag">li</span>>
 *span class="hijs-tag">

  <span class="hlis-tag"><<span class="hlis-name">Route</span> <span class="hlis-attr">path</span>=<span class="hlis-string">"\user/detail/:id"</span> <span class="hlis-attr">component</span>=<span</pre>
  class="hljs-string">{UserDetail}/</span>></span>
  <span class="hlis-tao"><<span class="hlis-name">Route</span><span class="hlis-attr">cmponent</span>=<span class="hlis-att
  string">{Post}/</span>></span>
  <span class="hljs-tag">div</span>>
```

```
<span class="hljs-tag">Router</span>>
  <span class="hljs-built_in">document</span>.getElementById(<span class="hljs-string">'root'</span</pre>
</code><h3 id="t8315.2 hooks.js">15.2 hooks.js <a href="#t8315.2 hooks.js">#t8315.2 hooks.js"<#t8315.2 hooks.js">#t8315.2 hooks.js"<#t8315.2 hooks.js">#t8315.2 hooks.js"<#t8315.2 hooks.js">#t8315.2 hooks.js"<#t8315.2 hooks.js"<#t8315.2 hooks.js">#t8315.2 hooks.js"<#t8315.2 hooks.js"<#t8315.2
 <span class="hljs-keyword">import</span> RouterContext <span class="hljs-keyword">from</span> <span class="hljs-string">'./RouterContext'</span>;
 <span class="hljs-keyword">import</span> matchPath <span class="hljs-keyword">from</span> <span class="hljs-string">'./matchPath'</span>
<span class="hljs-keyword">export</span> <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">useParams</span>()</span>{
<span class="hljs-keyword">let</span> match = React.useContext(RouterContext).match;
 <span class="hljs-keyword">return</span> match?match.params:{};
<span class="hljs-keyword">export</span> <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">useLocation</span>()</span>{<span class="hljs-keyword">retum</span> React.useContext(RouterContext).location;
 .
<span class="hljs-keyword">export</span> <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">useHistory</span>()</span>()</span>()
 <span class="hljs-keyword">return</span> React.useContext(RouterContext).history;
 <span class="hlis-keyword">export</span> <span class="hlis-function"><span class="hlis-function"><span class="hlis-keyword">function</span> <span class="hlis-title">useRouteMatch</span>(<span class="hlis-params">params">path</span></span></span></span></span></span></span></span></span>
 </span>{
 <soan class="hlis-keyword">const</soan> location = useLocation();
 <span class="hljs-keyword">let</span> match = React.useContext(RouterContext).match;
 <span class="hljs-keyword">return</span> path?matchPath(location.pathname,path):match;
 .</code><h3 id="t8415.3 react-routerlindex.js">15.3 react-routerlindex.js">15.3 react-routerlindex.js">#<h3 id="t8415.3 react-routerlindex.js">#
{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'
export (default as Prompt) from './Prompt':
 <span class="hljs-addition">+export {useHistory,useLocation,useParams,useRouteMatch} from './hooks';</span>
 </code><h2 id="t8516.路由懶加载">16.路由懶加载 <a href="#t8516.路由懶加载">#</a></h2><h3 id="t8616.1 src\index.js">16.1 src\index.js <a href="#t8616.1 src\index.js">#</a></h3>
 srclindex.js<code class="lang-js"><span class="hljs-keyword">import</pan> React, { Suspense } <span class="hljs-keyword">from</span> <span class="hljs-keymord">from</span> <span class="hljs-keymord"
 <soan class="hlis-keyword">import</soan> ReactDOM <soan class="hlis-keyword">from</soan> <soan class="hlis-string">react-dom'</soan>
 <span class="hljs-keyword">import</span> {
BrowserRouter <span class="hlis-keyword">as</span> Router, Route, Link
} <span class="hljs-keyword">from</span> <span class="hljs-string">'react-router-dom'</span>;
 <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">lazy</span>(<span class="hljs-params">load</span>) </span>[
<span class="hljs-keyword">returm</span> <span class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword">class="hljs-keyword"
state = { <span class="hljs-attr">InnerComponent</span>: <span class="hljs-literal">null</span> }
componentDidMount() {
load(),then(<span class="hljs-function"><span class="hljs-params">result</span> =></span> {
<span class="hljs-keyword">this</span>.setState({ <span class="hljs-attr">lnnerComponent</span>: result.default || result });
}):
render() {
 <span class="hljs-keyword">let</span> { InnerComponent } = <span class="hljs-keyword">this</span>.state;
 <span class="hljs-keyword">retum</span> InnerComponent? <span class="xml"><span class="hljs-tag"><<span class="hljs-name">InnerComponent</span> /></span></span></span> /></span class="hljs-name">InnerComponent</span> /></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></span></tp>
literal">null</span>;
 . <span class="hljs-keyword">const</span> LazyHome = React.lazy(<span class="hljs-function"><span class="hljs-params">()</span> =></span> <span class="hljs-keyword">import</span>(<span class="hljs-keyword")
string">'./components/Home'</span>));
 -span class="hijs-keyword">const-/span> LazyLogin = React.lazy(<span class="hijs-function"><span class="hijs-keyword">const-/span> <span class="hijs-keyword">const-/span> class="hijs-keyword">const-/span> <span class="hijs-keyword">const-/span> class="hijs-keyword">const-/span> <span class="hijs-keyword">const-/span> class="hijs-keyword">const-
string">'./components/Login'</span>));
 <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">Loading</span>() </span>{
<soan class="hlis-keyword">return/soan><soan class="mlis-tag">/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>/soan>
 <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">SuspenseHome</span>() </span>{
 <span class="hljs-keyword">return</span> (
 <span class="xml"><span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-attr">Loading</span> />
 </span>}>
<span class="hljs-tag"><<span class="hljs-name">LazyHome</span> /></span>
 <span class="hljs-tag">Suspense</span>></span>
 <span class="hljs-function"><span class="hljs-keyword">function</span> <span class="hljs-title">SuspenseLogin</span>() </span>{
 <span class="hljs-keyword">return</span> (
 <span class="xml"><span class="hijs-tag"><<span class="hijs-tag"><</p>
 </span>}>
 <span class="hljs-tag"><<span class="hljs-name">LazyLogin</span> /></span>
  <span class="hljs-tag">Suspense</span>></span>
ReactDOM.render(
 <span class="xml"><span class="hljs-tag"><<span class="hljs-name">Router</span>></span>
 <span class="hljs-tag"><<span class="hljs-name">div</span>></span>
 <span class="hljs-tag"><<span class="hljs-name">ul</span>></span>
 <span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"><<span class="hijs-tag"></span class="hijs-tag"></
</span>Home<span class="hljs-tag">Link</span>></span<span class="hljs-tag">li</span>>
<span class="hljs-tag"><<span class="hljs-tag">
 </span>></span>User<span class="hljs-tag">Link</span>><span class="hljs-tag">li</span>>
 <span class="hljs-tag">ul</span>>
 <span class="hijs-tag"><<span class="hijs-attr">pathsepan class="hijs-attr">exactsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">pathsepan class="hijs-attr">sepan class="hijs-attr">sep
attr">component</span>=<span class="hljs-string">{SuspenseHome}</span> /></span>
<pan class="hljs-tag"><<pan class="hljs-tag"><<pan class="hljs-tag"><<pan class="hljs-tag"></pan > <<pan class="hljs-tag">/login"</pan> <<pan class="hljs-tag"></pan > <<p>hljs-tag">/span > <<p>hljs-tag">/span > <<p>hljs-tag">/span > 
<span class="hljs-tag">div</span>>
<span class="hljs-tag">Router</span>>,
<span class="hljs-built_in">document</span>.getElementById(<span class="hljs-string">'root'</span>)
 </code><h2 id="t87链接">链接 <a href="#t87链接">#</a></h2><a href="mttps://gitee.com/zhufengpeixun/zhufeng_react_router_20201106_prepare">zhufeng_react_router_4a>
```