脚手架的搭建

1.npm install -g create-react-app

2.create-react-app 项目名字

路由 react-router2.x和3.x版本

安装 npm install -S react-router

使用

import {Router, Route, hashHistory, IndexRoute} from 'react-router'

<Router history = { hashHistory }>

<!-- 嵌套路由 -->

<Route path='/' component={myApp}>

<!-- 默认路由 -->

<IndexRoute component={defaultHome}/>

<Route path='products' component='productList' />

<Route path='personal' component='personalBox' />

</Route>

<!-- 并列路由 -->

<Route path='test' component='test'></Route>

</Router>

获取路由参数

this.props.params.paramName

路由的跳转(类似重定向)

现在访问/inbox/messages/5，会自动跳转到/messages/5

<Route path="inbox" component={Inbox}>

＜Redirect from="messages/:id" to="/messages/:id" />

</Route>

用户访问根路径时，将自动重定向到子组件welcome

<Route path="/" component={App}>

＜IndexRedirect to="/welcome" />

<Route path="welcome" component={Welcome} />

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

</Route>

标签中路由的跳转

<Link to="/about" activeStyle={{color: 'red'}} activeClassName="active">About</Link>

activeStyle 内联样式

activeClassName 类

跳转到根路由

<IndexLink to="/" activeClassName="active">

Home

</IndexLink>

js中的路由跳转

import { browserHistory } from 'react-router';

browserHistory.push('/some/path');

路由4.x版本

安装 npm install -S react-router-dom

使用

路由模式

//判断浏览器是否支持BrowserRouter模式

const supportsHistory = 'pushState' in window.history;

<BrowserRouter forceRefresh={!supportsHistory}> </BrowserRouter> history模式

<HashRouter></HashRouter> 哈希模式

<Switch>是独特的，因为它仅仅渲染一个路由

定义路由

<Route path="/news" component={News}></Route>

<Route path="/news" component={News} exact></Route> 严格匹配

路由跳转

<Link to="/about">跳转</Link> 跳转

<Link to="/about" replace>跳转</Link> 替换

<NavLink to="/about" 带有样式变化的跳转

activeClassName="selected" 激活的类

activeStyle={{ 激活的样式

fontWeight: 'bold',

color: 'red'

}}

>点击我会有样式改变</NavLink>

重定向

<Redirect to="/404"></Redirect>



