

Web开发技术

Web Application Development

第4课

WEB前端-REACT EXAMPLE

Episode Four

React Example

陈昊鹏

chen-hp@sjtu.edu.cn

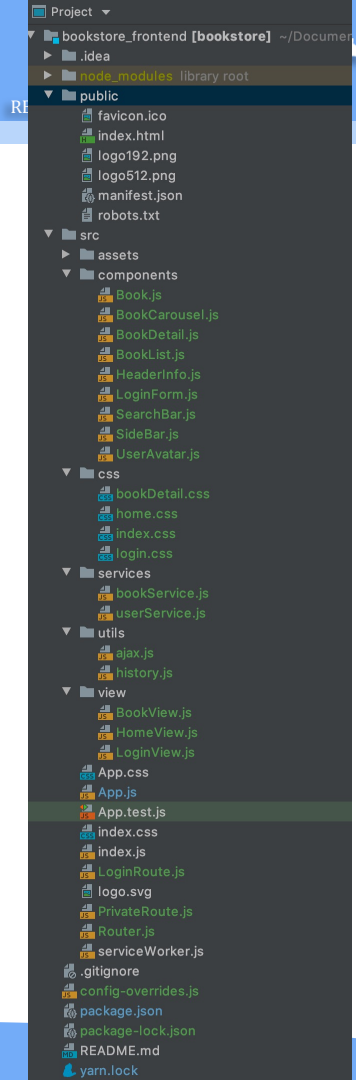


Web Application
Development

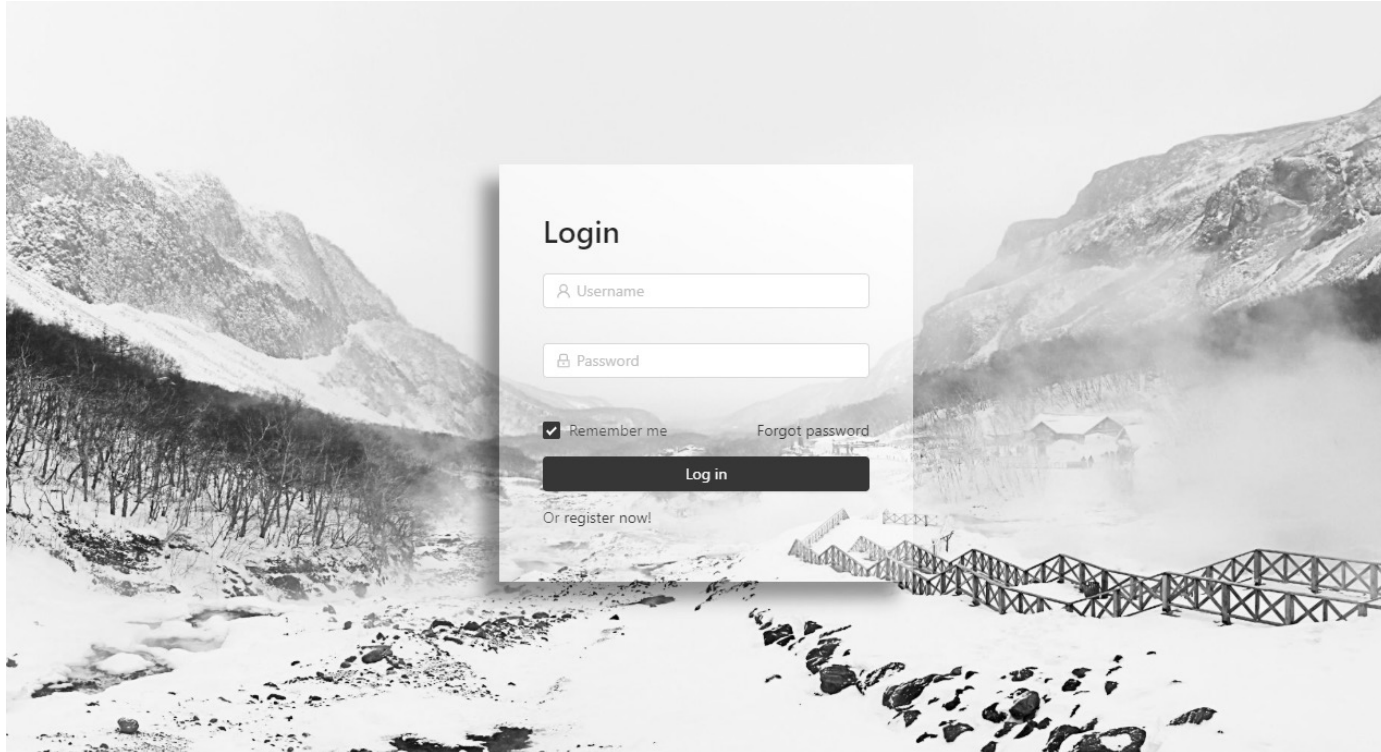
- Frontend
 - React
 - HTML/CSS/JS
 - WebStorm
 - Ant Design
- Backend
 - SSS (Spring + SpringMVC + Spring Data JPA)
 - Java
 - IntelliJ IDEA

An Optional Architecture

- assets
 - images, documents, ...
- components
 - React components
- CSS
 - CSS files
- services
 - some functions which provide services
- utils
 - help functions
- view
 - React views



CSS is an important factor to make pages look more beautiful.



```
class LoginView extends React.Component{

  render(){
    return(
      <div className="login-page">
        <div className="login-container">
          <div className="login-box">
            <h1 className="page-title">Login</h1>
            <div className="login-content">
              <WrappedLoginForm />
            </div>
          </div>
        </div>
      </div>
    );
  }
}
```

We use the component designed by Ant Design as an example.

```
class LoginForm extends React.Component {
  const { getFieldDecorator } = this.props.form;
  render() {
    return (
      <Form onSubmit={this.handleSubmit} className="login-form">
        <Form.Item> /* form element here */ </Form.Item>
        <Form.Item> /* form element here */ </Form.Item>
        ...
        <Form.Item>
          <Button type="primary" htmlType="submit" className="login-form-button">
            Log in
          </Button>
          Or <a href="">register now!</a>
        </Form.Item>
      </Form>
    );
  }
}

const WrappedLoginForm = Form.create({ name: 'normal_login' })(LoginForm);
export default WrappedLoginForm
```

```
<Form.Item>
  {getFieldDecorator('username', {
    rules: [{ required: true, message: 'Please input your username!' }],
  })(
    <Input
      prefix={<Icon type="user"/>}
      placeholder="Username"
    />,
  )}
</Form.Item>
<Form.Item>
  {getFieldDecorator('password', {
    rules: [{ required: true, message: 'Please input your Password!' }],
  })(
    <Input
      prefix={<Icon type="lock"/>}
      type="password"
      placeholder="Password"
    />,
  )}
</Form.Item>
```

```
handleSubmit = e => {
  e.preventDefault();
  this.props.form.validateFields((err, values) => {
    if (!err) {
      console.log('Received values of form: ', values);
      userService.login(values);
    }
  });
};

/* In userService.js */
export const login = (data) => {
  const url = `${config.apiUrl}/login`; /* get request url */
  const callback = (data) => {
    if(data.status >= 0) {
      localStorage.setItem('user', JSON.stringify(data.data)); /* save user info */
      history.push("/"); /* use react-router-dom here to redirect to the web page */
      message.success(data.msg); /* message box */
    }
    else{
      message.error(data.msg);
    }
  };
  postRequest(url, data, callback); /* ajax */
};
```


- Traditional Concept: Ajax
- Sending requests to specified API to get responses
- Usually in JSON format
- Fetch / Axios

```
/* using fetch as an example */
let postRequest = (url, json, callback) => {

  let opts = {
    method: "POST",
    body: JSON.stringify(json),
    headers: {
      'Content-Type': 'application/json'
    },
    credentials: "include"    /* take cookies in fetch API */
  };

  fetch(url,opts)
    .then((response) => {
      return response.json()
    })
    .then((data) => {
      callback(data);
    })
    .catch((error) => {
      console.log(error);
    });
};
```

Backend Controller (@RequestBody)



Book Store

Hi, thunderboy



Books

My Cart

My Orders

My Profile

input here



Java核心技术卷II

¥95.2



深入理解计算机系统

¥136.9



Effective C++

¥51.3



小王子

¥8.89



```
class HomeView extends React.Component{

  constructor(props) {
    super(props);

  }

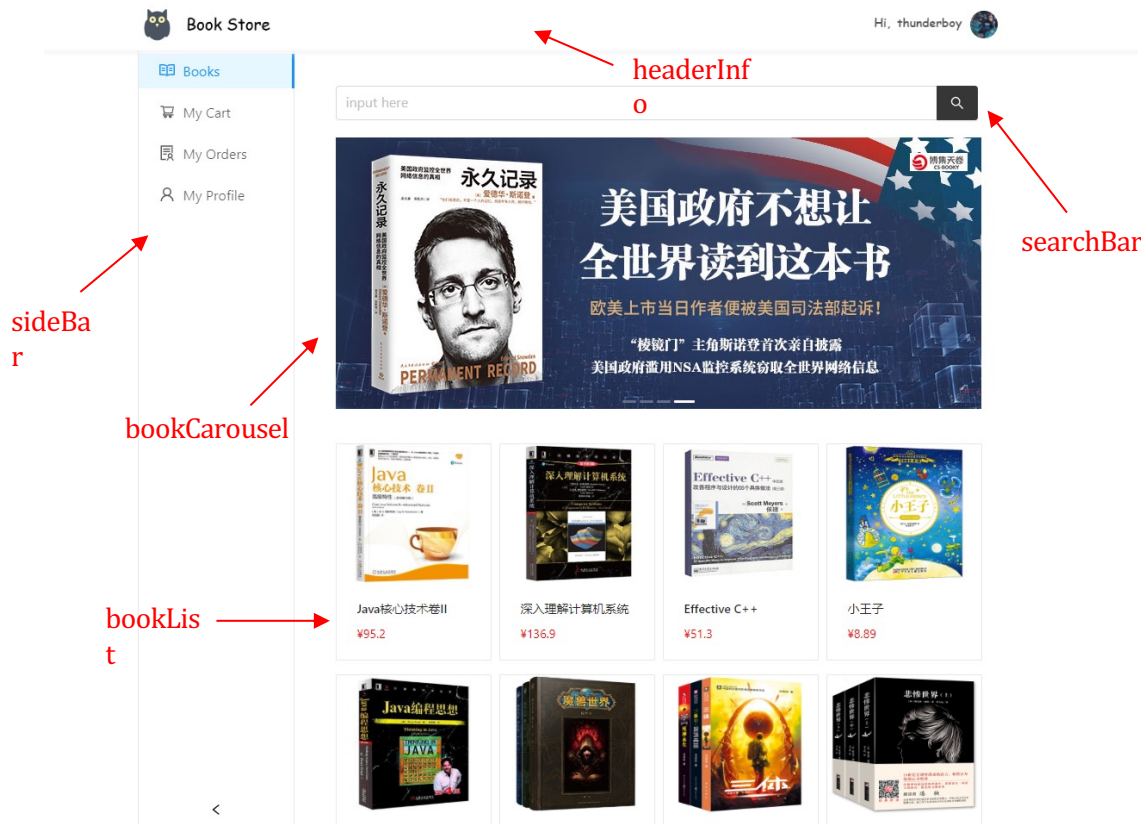
  componentDidMount(){ /* get the user info when the page has first rendered */
    let user = localStorage.getItem("user");
    this.setState({user:user});
  }

  render(){...}

}
```

```
import {Layout} from 'antd';
import ...

const { Header, Content} = Layout;
render(){
  return(
    <Layout className="layout">
      <Header>
        <HeaderInfo />
      </Header>
      <Layout>
        <SideBar />
        <Content style={{ padding: '0 50px' }}>
          <div className="home-content">
            <SearchBar />
            <BookCarousel />
            <BookList />
          </div>
        </Content>
      </Layout>
    </Layout>
  );
}
```



```
import React from 'react';
import {List} from 'antd';
import {Book} from './Book';
import {getBooks} from '../services/bookService';

export class BookList extends React.Component{

  constructor(props) {
    super(props);
    this.state = {books:[]};
  }

  componentDidMount() {

    const callback = (data) => {
      this.setState({books:data});
    };

    getBooks({"search":null}, callback);  /* fetch books from backend */
  }

  render() {...}

}
```

```
/* In bookService.js */

import {postRequest} from "../utils/ajax";

/* get book list */
export const getBooks = (data, callback) => {
  const url = `${config.apiUrl}/getBooks`;
  postRequest(url, data, callback);
};

/* get single book info by a specified id */
export const getBook = (id, callback) => {
  const data = {id: id};
  const url = `${config.apiUrl}/getBook`;
  postRequest(url, data, callback);
};
```



```
render() {  
  return (  
    <List  
      grid={{gutter: 10, column: 4}}           /* 4 books in one row */  
      dataSource={this.state.books}  
      pagination={{  
        onChange: page => {pageSize: 16}}      /* 16 books in one page */  
        renderItem={item => (  
          <List.Item>  
            <Book info={item} />  
          </List.Item>  
        )}  
      />  
    );  
  }  
}
```

```
import React from 'react';
import { Card } from 'antd';
import { Link } from 'react-router-dom';

const { Meta } = Card;
export class Book extends React.Component{
  render() {
    const {info} = this.props;
    return (
      <Link to={{          /* use Link to open a new page about detailed book info */
        pathname: '/bookDetails',
        search: '?id=' + info.bookId}}
        target="_blank">
        <Card
          hoverable
          cover={<img alt="image" src={info.image} className={"bookImg"}/>}
        >
          <Meta title={info.name} description={'¥' + info.price}/>
        </Card>
      </Link>
    );
  }
}
```

 Book Store

Hi, thunderboy

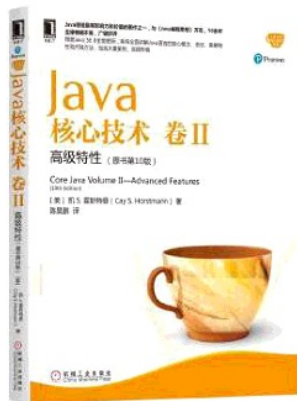


 Books

 My Cart

 My Orders

 My Profile



Java核心技术卷II

作者：凯S.霍斯特曼

分类：编程

定价：¥95.2

状态：有货 库存1000件

作品简介：

本书是Java领域有影响力 and 价值的著作之一，由拥有20多年教学与研究经验的Java技术专家撰写（获Jolt大奖），与《Java编程思想》齐名，10余年全球畅销不衰，广受好评。第10版根据JavaSE8全面更新，同时修正了第9版中的不足，系统全面讲解了Java语言的核心概念、语法、重要特性和开发方法，包含大量案例，实践性强。

 加入购物车

 立即购买

```
import {getBook} from "../services/bookService";
import {BookDetail} from "../components/BookDetail";

class BookView extends React.Component{

  constructor(props) {
    super(props);
    this.state = {books:null};
  }

  componentDidMount(){
    let user = localStorage.getItem("user");
    this.setState({user:user});

    const query = this.props.location.search; /* url: e.g. localhost:3000/bookDetails?id=1 */
    const arr = query.split('&');
    const bookId = arr[0].substr(4); /* acquire the book id */
    this.setState({bookId:bookId});
    getBook(bookId, (data) => {this.setState({bookInfo: data})}) /* get the detailed info */
  }

  render(){...}
}
```

```
render(){
  return(
    <Layout className="layout">
      <Header>
        <HeaderInfo />      /* head bar */
      </Header>
      <Layout>
        <SideBar />          /* side bar */
        <Content style={{ padding: '0 50px' }}>
          <div className="home-content">
            <BookDetail info={this.state.bookInfo} /> /* detailed book info */
          </div>
        </Content>
      </Layout>
    </Layout>
  );
}
```

```
import { Descriptions, Button } from 'antd';

export class BookDetail extends React.Component{
  render() {
    const {info} = this.props;
    if(info == null){ return null; }    /* if the data is not available, return null instead */
    return (
      <div className={"content"}>
        <div className={"book-detail"}>
          <div className={"book-image"}><img alt="image" src={info.image} /></div>
          <div className={"descriptions"}>
            <Descriptions> /* descriptions here */ </Descriptions>
          </div>
        </div>
        <div className={"button-groups"}> /* buttons */
          <Button type="danger" icon="shopping-cart" size={"large"}>加入购物车</Button>
          <Button type="danger" icon="pay-circle" size={"large"} ghost>立即购买</Button>
        </div>
      </div>
    )
  }
}
```

```
import { Descriptions } from 'antd';
```

```
<Descriptions>
```

```
  <Descriptions.Item className={"title"} span={3}>{info.name}</Descriptions.Item>
```

```
  <Descriptions.Item label={"作者"} span={3}>{info.author}</Descriptions.Item>
```

```
  <Descriptions.Item label={"分类"} span={3}>{info.type}</Descriptions.Item>
```

```
  <Descriptions.Item label={"定价"} span={3}>
```

```
    {<span className={"price"}>{'¥' + info.price}</span>}
```

```
  </Descriptions.Item>
```

```
  <Descriptions.Item label={"状态"} span={3}>
```

```
    {info.inventory !== 0? <span>有货 库存{info.inventory}件</span>:<span>无货</span>}
```

```
  </Descriptions.Item>
```

```
  <Descriptions.Item label={"作品简介"} span={3}>{info.description}</Descriptions.Item>
```

```
</Descriptions>
```

- What is a Router?
 - A router allows your application to navigate between different components, changing the browser URL, modifying the browser history, and keeping the UI state in sync.
- React Router
 - the most popular routing library for React
- Packages
 - react-router: core package for the router
 - **react-router-dom**: the router components for websites
 - react-router-native: for an app development environment (React Native)

- React Router API is based on three components:
 - `<Router>`: The router that keeps the UI in sync with the URL
 - `<Link>`: Renders a navigation link
 - `<Route>`: Renders a UI component depending on the URL
- `<Router>`
 - `<BrowserRouter>`
 - uses the HTML5 History API
 - uses regular URL paths (best-looking)
 - requires your server to be configured correctly
 - `<HashRouter>`
 - uses the hash portion of the URL
 - uses hash-style URL paths e.g. `http://example.com/#/your/page`
 - no special server configuration is needed

- `<Switch>` & `<Route>`

- when a `<Switch>` is rendered, it searches through its children `<Route>` elements to find one whose path matches the current URL
- when it finds one, it renders that `<Route>` and ignores all others
- if no `<Route>` matches, the `<Switch>` renders nothing (null)

```
render(){  
  return(  
    <Router>  
      <Switch>  
        <Route path="/about"><About /></Route>  
        <Route path="/users"><Users /></Route>  
        <Route path="/"><Home /></Route>  
      </Switch>  
    </Router>  
  )  
}
```

- **<Link>**
 - creates links in your application
 - wherever you render a **<Link>**, an anchor (**<a>**) will be rendered in your HTML document

```
<Link to="/">Home</Link>  
// <a href="/">Home</a>
```

- **<Redirect>**
 - forces navigation
 - when a **<Redirect>** renders, it will navigate using its to prop

```
<Redirect to="/login" />
```

A Simple Example

```
import React from "react";
import { BrowserRouter as Router, Switch, Route, Link } from "react-router-dom";

export default function App() {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li><Link to="/">Home</Link></li>
            <li><Link to="/about">About</Link></li>
            <li><Link to="/users">Users</Link></li>
          </ul>
        </nav>
        <Switch>
          <Route path="/about"><About /></Route>
          <Route path="/users"><Users /></Route>
          <Route path="/"><Home /></Route>
        </Switch>
      </div>
    </Router>
  );
}
```

```
function Home() {
  return <h2>Home</h2>;
}

function About() {
  return <h2>About</h2>;
}

function Users() {
  return <h2>Users</h2>;
}
```

Router.js

```
import React from 'react';
import {Router, Route, Switch} from 'react-router-dom';
import HomeView from '../HomeView';
import LoginView from '../LoginView';
import BookView from '../BookView';

const BasicRoute = () => (
  <Router>
    <Switch>
      <Route exact path="/" component={HomeView}/>
      <Route exact path="/login" component={LoginView}/>
      <Route exact path="/bookDetails" component={BookView}/>
    </Switch>
  </Router>
);

export default BasicRoute;
```

App.js

```
import React from 'react';
import './App.css';
import BasicRoute from "../Router";

class App extends React.Component {

  render() {
    return (
      <BasicRoute/>
    );
  }
}

export default App;
```

```
import React from 'react';
import {Route, Redirect} from 'react-router-dom';
import * as userService from './services/userService';
import {message} from 'antd';

export default class PrivateRoute extends React.Component{

  constructor(props) {
    super(props);
    this.state = {
      isAuthenticated: false,
      hasAuthenticated: false,
    };
  }

  componentDidMount() {
    userService.checkSession(this.checkAuth);
  }

  render() {...}
}
```

```
const checkSession = (callback) => {  
  const url = `${config.apiUrl}/checkSession`;  
  postRequest(url, {}, callback);  
};  
  
checkAuth = (data) => {  
  if (data.status >= 0) {  
    this.setState({isAuthenticated: true, hasAuthenticated: true});  
  }  
  else {  
    message.error(data.msg);  
    this.setState({isAuthenticated: false, hasAuthenticated: true});  
  }  
};
```



```
render() {  
  
  const {component: Component, path="/", exact=false, strict=false} = this.props;  
  
  if (!this.state.hasAuthenticated) {  
    return null;  
  }  
  
  return <Route path={path} exact={exact} strict={strict} render={props => (  
    this.state.isAuthenticated ? (  
      <Component {...props}/>  
    ) : (  
      /* if the user has not authenticated, redirect to the login page */  
      <Redirect to={{  
        pathname: '/login',  
        state: {from: props.location}  
      }}/>  
    )  
  )}/>  
}
```

Router.js

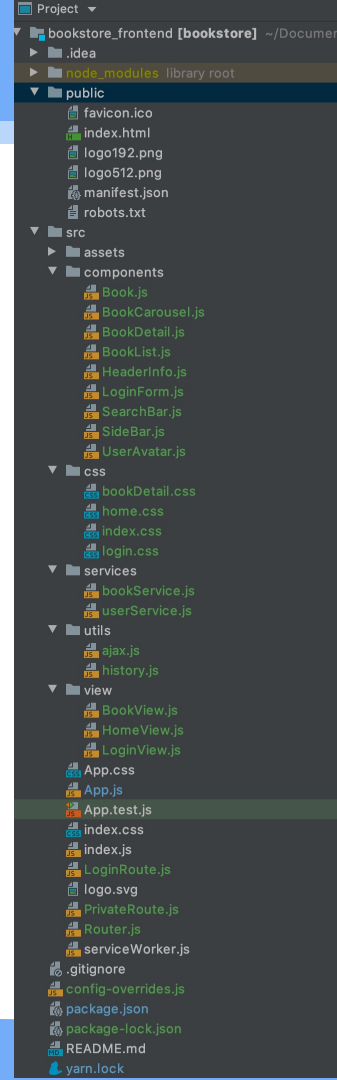
```
import React from 'react';
import {Router, Route, Switch} from 'react-router-dom';
import HomeView from '../HomeView';
import LoginView from '../LoginView';
import BookView from '../BookView';
import PrivateRoute from './PrivateRoute';

const BasicRoute = () => (
  <Router>
    <Switch>
      <PrivateRoute exact path="/" component={HomeView}/>
      <Route exact path="/login" component={LoginView}/>
      <PrivateRoute exact path="/bookDetails" component={BookView}/>
    </Switch>
  </Router>
);

export default BasicRoute;
```

An Optional Architecture

- assets
 - images, documents, ...
- components
 - React components
- CSS
 - css files
- services
 - some functions which provide services
- utils
 - help functions
- view
 - React views



```
import React from 'react';  
import ReactDOM from 'react-dom';  
import './index.css';  
import App from './App';  
import * as serviceWorker from './serviceWorker';
```

```
ReactDOM.render(<App />, document.getElementById('root'));
```

```
// If you want your app to work offline and load faster, you can change  
// unregister() to register() below. Note this comes with some pitfalls.  
// Learn more about service workers: https://bit.ly/CRA-PWA  
serviceWorker.unregister();
```

```
import React from 'react';  
import './App.css';  
import BasicRoute from './Router';  
  
class App extends React.Component {  
  
  render() {  
    return (  
      <BasicRoute/>  
    );  
  }  
}  
  
export default App;
```

```
import React from 'react';  
import { Router, Route, Switch, Redirect } from 'react-router-dom';  
import PrivateRoute from './PrivateRoute'  
import LoginRoute from './LoginRoute'  
import HomeView from './view/HomeView';  
import LoginView from './view/LoginView'  
import { history } from './utils/history';  
import BookView from './view/BookView';
```

```
class BasicRoute extends React.Component{  
  
  constructor(props) {  
    super(props);  
  
    history.listen((location, action) => {  
      // clear alert on location change  
      console.log(location, action);  
    });  
  }  
}
```

```
render() {  
  return(  
    <Router history={history}>  
      <Switch>  
        <PrivateRoute exact path="/" component={HomeView} />  
        <LoginRoute exact path="/login" component={LoginView} />  
        <PrivateRoute exact path="/bookDetails" component={BookView} />  
        <Redirect from="/*" to="/" />  
      </Switch>  
    </Router>  
  )  
}  
  
}  
  
export default BasicRoute;
```

```
import { createBrowserHistory } from 'history';
```

```
export const history = createBrowserHistory();
```

- The history library provides history tracking and navigation primitives for JavaScript applications that run in browsers and other stateful environments.
- We provide 3 different methods for working with history, depending on your environment:
 - A "browser history" is for use in modern web browsers that support the [HTML5 history API](#)
 - A "hash history" is for use in web browsers where you want to store the location in the [hash](#) portion of the current URL to avoid sending it to the server when the page reloads
 - A "memory history" is used as a reference implementation that may be used in non-browser environments, like [React Native](#) or tests
 - From: <https://github.com/ReactTraining/history/blob/master/docs/getting-started.md>

- **Properties**

- Each history object has the following properties:
- [history.location](#) - The current location
- [history.action](#) - The current navigation action
- Additionally, memory history provides **history.index** that tells you the current index in the history stack.

- **Listening**

- You can listen for changes to the current location using **history.listen**:

```
history.listen(({ action, location }) => {  
  console.log(  
    `The current URL is ${location.pathname}${location.search}${location.hash}`  
  );  
  console.log(`The last navigation action was ${action}`);  
});
```

- **From:** <https://github.com/ReactTraining/history/blob/master/docs/getting-started.md>

```
import React from 'react';
import {Route, Redirect} from 'react-router-dom'
import * as userService from "../services/userService"
import {message} from "antd";
```

```
export class LoginRoute extends React.Component{
  constructor(props) {
    super(props);
    this.state = {
      isAuthenticated: false,
      hasAuthenticated: false,
    };
  }

  checkAuth = (data) => {
    console.log(data);
    if (data.status >= 0) {
      this.setState({isAuthenticated: true, hasAuthenticated: true});
    } else {
      localStorage.removeItem('user');
      this.setState({isAuthenticated: false, hasAuthenticated: true});
    }
  };
}
```

```
componentDidMount() {
  userService.checkSession(this.checkAuth);
}

render() {
  const {component: Component, path="/", exact=false, strict=false} = this.props;

  console.log(this.state.isAuthenticated);

  if (!this.state.isAuthenticated) {
    return null;
  }

  return <Route path={path} exact={exact} strict={strict} render={props => (
    this.state.isAuthenticated ? (
      <Redirect to={{
        pathname: '/',
        state: {from: props.location}
      }}/>
    ) : (
      <Component {...props}/>
    )
  )}/>
}

export default LoginRoute
```

```
import React from 'react';
import {Route, Redirect} from 'react-router-dom'
import * as userService from "../services/userService"
import {message} from "antd";

export default class PrivateRoute extends React.Component{
  constructor(props) {
    super(props);
    this.state = {
      isAuthenticated: false,
      hasAuthenticated: false,
    };
  }

  checkAuth = (data) => {
    console.log(data);
    if (data.status >= 0) {
      this.setState({isAuthenticated: true, hasAuthenticated: true});
    } else {
      message.error(data.msg);
      localStorage.removeItem('user');
      this.setState({isAuthenticated: false, hasAuthenticated: true});
    }
  };
}
```

```
componentDidMount() {
  userService.checkSession(this.checkAuth);
}

render() {
  const {component: Component, path="/", exact=false, strict=false} = this.props;

  console.log(this.state.isAuthenticated);

  if (!this.state.isAuthenticated) {
    return null;
  }

  return <Route path={path} exact={exact} strict={strict} render={props => (
    this.state.isAuthenticated ? (
      <Component {...props}/>
    ) : (
      <Redirect to={{
        pathname: '/login',
        state: {from: props.location}
      }}/>
    )
  )} />
}
```

```
import React from 'react';
import {Layout, Carousel} from 'antd'
import {HeaderInfo} from "../components/HeaderInfo";
import {SideBar} from "../components/SideBar";
import '../css/home.css'
import {withRouter} from "react-router-dom";
import {BookCarousel} from "../components/BookCarousel";
import {SearchBar} from "../components/SearchBar";
import {BookList} from "../components/BookList";
```

```
const { Header, Content, Footer } = Layout;
```

```
class HomeView extends React.Component{
```

```
  constructor(props) {
    super(props);
```

```
  }
```

```
  componentDidMount(){
    let user = localStorage.getItem("user");
    this.setState({user:user});
  }
```

```
  render(){
    return(
      <Layout className="layout">

        <Header>
          <HeaderInfo />
        </Header>
        <Layout>
          <SideBar />
          <Content style={{ padding: '0 50px' }}>
            <div className="home-content">
              <SearchBar />

              <BookCarousel />
              <BookList />
              <div className="foot-wrapper">
                </div>
            </div>
          </Content>
        </Layout>
      </Layout>
    );
  }
}
```

```
export default withRouter(HomeView);
```

```
import config from 'config';
import {postRequest} from "../utils/ajax";
import {history} from '../utils/history';
import {message} from 'antd';
```

```
export const login = (data) => {
  const url = `${config.apiUrl}/login`;
  const callback = (data) => {
    if(data.status >= 0) {
      localStorage.setItem('user', JSON.stringify(data.data));
      history.push("/");
      message.success(data.msg);
    }
    else{
      message.error(data.msg);
    }
  };
  postRequest(url, data, callback);
};
```

```
export const logout = () => {
  const url = `${config.apiUrl}/logout`;

  const callback = (data) => {
    if(data.status >= 0) {
      localStorage.removeItem("user");
      history.push("/login");
      message.success(data.msg);
    }
    else{
      message.error(data.msg);
    }
  };
  postRequest(url, {}, callback);
};

export const checkSession = (callback) => {
  const url = `${config.apiUrl}/checkSession`;
  postRequest(url, {}, callback);
};
```

```
import React from 'react';  
import WrappedLoginForm from '../components/LoginForm';  
import {withRouter} from "react-router-dom";
```

```
class LoginView extends React.Component{
```

```
  render(){  
    return(  
      <div className="login-page">  
        <div className="login-container">  
          <div className="login-box">  
            <h1 className="page-title">Login</h1>  
            <div className="login-content">  
              <WrappedLoginForm />  
            </div>  
          </div>  
        </div>  
      </div>  
    );  
  }  
}
```

```
export default withRouter(LoginView);
```



```
import React from 'react';  
import { Form, Icon, Input, Button, Checkbox } from 'antd';  
import 'antd/dist/antd.css';  
import '../css/login.css'  
import * as userService from '../services/userService'
```

```
class LoginForm extends React.Component {
```

```
  handleSubmit = e => {  
    e.preventDefault();  
    this.props.form.validateFields((err, values) => {  
      if (!err) {  
        console.log('Received values of form: ', values);  
        userService.login(values);  
      }  
    });  
  };  
};
```

```
render() {  
  const { getFieldDecorator } = this.props.form;  
  return (  
    <Form onSubmit={this.handleSubmit} className="login-form">  
      <Form.Item>  
        {getFieldDecorator('username', {  
          rules: [{ required: true, message: 'Please input your username!' }],  
        })(  
          <Input  
            prefix={<Icon type="user" style={{ color: 'rgba(0,0,0,.25)' }} />}  
            placeholder="Username"  
          />,  
        )}  
      </Form.Item>  
      <Form.Item>  
        {getFieldDecorator('password', {  
          rules: [{ required: true, message: 'Please input your Password!' }],  
        })(  
          <Input  
            prefix={<Icon type="lock" style={{ color: 'rgba(0,0,0,.25)' }} />}  
            type="password"  
            placeholder="Password"  
          />,  
        )}  
      </Form.Item>  
    </Form>  
  );  
}
```

```
<Form.Item>
  {getFieldDecorator('remember', {
    valuePropName: 'checked',
    initialValue: true,
  })(<Checkbox>Remember me</Checkbox>)}
  <a className="login-form-forgot" href="">
    Forgot password
  </a>
  <Button type="primary" htmlType="submit" className="login-form-button">
    Log in
  </Button>
  Or <a href="">register now!</a>
</Form.Item>
</Form>
);
}
}

const WrappedLoginForm = Form.create({ name: 'normal_login' })(LoginForm);

export default WrappedLoginForm
```

- Ant Design of React
- <https://ant.design/docs/react/introduce-cn>
- React Router 中文文档
- <http://react-guide.github.io/react-router-cn/index.html>
- React Router 使用教程
- <https://www.jianshu.com/p/78d551b6d1ac>
- 在 React Router 4 中建立具有權限檢核的 Private Route 組件
- <https://dotblogs.com.tw/wasichris/2018/01/22/002904>
- React-Router v4 简单入门教程
- <https://www.jianshu.com/p/5e8297858ea8>
- React Router Tutorial
- <https://www.codingame.com/playgrounds/6517/react-router-tutorial>



- *Web*开发技术
- *Web Application Development*

Thank You!