

link: null  
title: 珠峰架构师成长计划  
description: 页面上的内容是由服务器生产的  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=411 sentences=1023, words=6590

## 1. 什么是服务器端渲染

页面上的内容是由服务器生产的

```
cnpm i express -S
```

```
let express=require('express');
let app=express();
app.get('/',(req,res) => {
  res.send(`
    <h1>hello</h1>
  `);
});
app.listen(8080);
```

页面上的内容由于浏览器运行JS脚本而渲染到页面上的

- 浏览器访问服务器
- 服务器返回一个空的HTML页面，里面有一个JS资源链接，比如bundle.js
- 浏览器下载JS代码并在浏览器中运行
- 内容呈现在页面上

```
cnpm i react react-dom -S
cnpm i webpack webpack-cli -D
cnpm i babel-loader @babel/core -D
cnpm i @babel/preset-env @babel/preset-react -D
cnpm i webpack-node-externals -D
```

```
let express=require('express');
let app=express();
app.get('/',(req,res) => {
  res.send(`
    <div id="root">
      <script src="/bundle.js"></script>
    </div>
  `);
});
app.listen(9090);
```

## 2. 服务器端打包React组件

- 1. 浏览器发送请求
- 1. 服务器运行React代码生成页面
- 1. 服务器返回页面

server/webpack.config.js

```
let path = require('path');
let nodeExternals=require('webpack-node-externals');
module.exports = {
  target: 'node',
  mode: 'development',
  output:{
    path:path.resolve(__dirname, 'build'),
    filename:'bundle.js'
  },
  externals:[nodeExternals()],
  module:{
    rules:[
      {
        test:/\.js$/,
        loader: 'babel-loader',
        exclude: /node_modules/,
        options: {
          presets: [
            [
              "@babel/preset-env", {
                targets: {
                  browsers:['last 2 versions']
                }
              },
              "@babel/preset-react"
            ]
          ]
        }
      }
    ]
  }
}
```

server/src/index.js

```
import Home from './containers/Home';
let express=require('express');
let app=express();
app.get('/', (req,res) => {
    res.send(`

        document.getElementById('root').innerHTML = 'hello';

    `);
});
app.listen(9090);
```

server/src/containers/Home/index.js

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

### 3. 服务器端渲染

server/webpack.config.js

```
let path = require('path');
let nodeExternals=require('webpack-node-externals');
module.exports = {
    target: 'node',
    mode: 'development',
    entry:path.resolve(__dirname,'./src/index.js'),
    output:{
        path:path.resolve(__dirname,'build'),
        filename:'bundle.js'
    },
    externals:[nodeExternals()],
    module:{
        rules:[
            {
                test: /\.js$/,
                loader: 'babel-loader',
                exclude: /node_modules/,
                options: {
                    presets: [
                        "@babel/preset-env",
                        "@babel/preset-react"
                    ]
                }
            }
        ]
    }
}
```

server/src/index.js

```
import React from 'react';
import Home from './containers/Home';
import {renderToString} from 'react-dom/server';
import express from 'express';
let app=express();
const html=renderToString(<Home />);
app.get('/', (req,res) => {
    res.send(`

        ${html}

    `);
});
app.listen(9090);
```

package.json

```
"scripts": {
    "start": "node ./server/build/bundle.js",
    "build":"webpack --config server/webpack.config.js"
}
```

### 4. 优化启动流程

cnpm i nodemon -g

```
"scripts": {
    "dev":"npm-run-all --parallel dev:**",
    "dev:start":"nodemon './server/build/bundle.js'",
    "dev:build":"webpack --config server/webpack.config.js --watch"
}
```

cnpm npm-run-all -g

### 5. 计数器组件

- 服务器端运行React代码渲染出HTML
- 服务器把渲染出的HTML页面发送给了浏览器
- 浏览器接受到HTML会渲染到页面上
- 浏览器发现页面引用的index.js文件会去下载
- 浏览器下载得到的index.js文件并在浏览器端执行
- 浏览器中的代码接管了页面的所有内容，后面和客户端渲染是一样的

src/client/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import Counter from '../containers/Counter';

ReactDOM.hydrate(<Counter/>, document.querySelector('#root'));
```

src/containers/Counter/index.js

```
import React, {Component} from 'react';
export default class Counter extends Component {
  state={number:0}
  render() {
    return (
      <div>
        <p>{this.state.number}</p>
        <button onClick={()=>this.setState({number:this.state.number+1})}>+button</button>
      </div>
    )
  }
}
```

webpack.client.js

```
let path = require('path');
module.exports = {
  mode: 'development',
  entry: path.resolve(__dirname, './src/client/index.js'),
  output: {
    path: path.resolve(__dirname, 'public'),
    filename: 'index.js'
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        loader: 'babel-loader',
        exclude: /node_modules/,
        options: {
          presets: [
            '@babel/preset-env',
            '@babel/preset-react'
          ],
          plugins: [
            '@babel/plugin-proposal-class-properties'
          ]
        }
      }
    ]
  }
}
```

## 6. 优化代码结构

```
let path = require('path');
module.exports = {
  mode: 'development',
  module: {
    rules: [
      {
        test: /\.js$/,
        loader: 'babel-loader',
        exclude: /node_modules/,
        options: {
          presets: [
            '@babel/preset-env',
            '@babel/preset-react'
          ],
          plugins: [
            '@babel/plugin-proposal-class-properties'
          ]
        }
      }
    ]
  }
}
```

webpack.client.js

```
let path=require('path');
const merge=require('webpack-merge');
const base=require('./webpack.base');
module.exports = merge(base, {
  mode: 'development',
  entry: path.resolve(__dirname, './src/client/index.js'),
  output: {
    path: path.resolve(__dirname, 'public'),
    filename: 'index.js'
  }
})
```

webpack.server.js

```

let path=require('path');
const merge=require('webpack-merge');
const base=require('./webpack.base');
const nodeExternals=require('webpack-node-externals');
module.exports = merge(base, {
  target: 'node',
  mode: 'development',
  entry:path.resolve(__dirname, './src/server/index.js'),
  output:{
    path:path.resolve(__dirname, 'build'),
    filename:'bundle.js'
  },
  externals:[nodeExternals()]
})

```

src/server/index.js

```

import React from 'react';
import {renderToString} from 'react-dom/server';
import Home from '../containers/Home';
import express from 'express';
let app=express();
app.use(express.static('public'));
const content=renderToString(<Home />);
app.get('/', (req, res) => {
  res.send(`
    <div>
      <h1>Hello World!</h1>
    </div>
  `);
});
app.listen(9090);

```

## 7. 使用路由

```
cnpm i react-router-dom -S
```

- 客户端请求服务器
- 服务器返回HTML给浏览器，浏览器渲染显示页面
- 浏览器发现需要外链JS资源，加载JS资源
- 加载好的JS资源在浏览器端执行
- JS中的React代码开始实现路由功能
- 路由代码首先获取地址栏中的地址，然后根据不同的地址根据路由配置渲染对应内容

src/client/index.js

```

import React from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import routes from '../routes';
ReactDOM.hydrate(
  <BrowserRouter>
    {routes}
  </BrowserRouter>
, document.querySelector('#root'));

```

src/server/index.js

```

import React from 'react';
import {renderToString} from 'react-dom/server';
import Home from '../containers/Home';
import express from 'express';
import {StaticRouter} from 'react-router-dom';
import routes from '../routes';
let app=express();
app.use(express.static('public'));

app.get('*', (req, res) => {
  const content=renderToString(
    <StaticRouter context={[]} location={req.path}>
      {routes}
    </StaticRouter>
  );
  res.send(`
    <div>
      <h1>Hello World!</h1>
    </div>
  `);
});
app.listen(9090);

```

src/routes.js

```

import React, {Fragment} from 'react';
import {Route} from 'react-router-dom';
import Home from '../containers/Home';
import Counter from '../containers/Counter';
export default (
  <Fragment>
    <Route path="/" exact component={Home}>Route</Route>
    <Route path="/counter" exact component={Counter}>Route</Route>
  </Fragment>
)

```

## 8. 跳转路由

src/client/index.js

```
import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';

ReactDOM.hydrate(
  <BrowserRouter>
    <Fragment>
      <Header/>
      <div className="container" style={{marginTop:50}}>
        {routes}
      </div>
    </Fragment>
  </BrowserRouter>
, document.querySelector('#root'));
```

src/containers/Counter/index.js

```
import React, {Component} from 'react';
export default class Counter extends Component {
  state={number:0}
  render() {
    return (
      <div>
        <p>{this.state.number}</p>
        <button className="btn btn-primary" onClick={()=>this.setState({number:this.state.number+1})}>+button</button>
      </div>
    )
  }
}
```

src/containers/Home/index.js

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

src/server/index.js

```
import express from 'express';
import render from './render';
let app=express();
app.use(express.static('public'));

app.get('*', (req, res) => {
  render(req, res);
});
app.listen(9090);
```

src/components/Header/index.js

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
export default class Home extends Component {
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">HomeLink</li>
              <li><Link to="/counter">CounterLink</li>
            </ul>
          </div>
        </div>
      </nav>
    )
  }
}
```

src/server/render.js

```
import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
export default function (req, res) {
  const content=renderToString(
    <StaticRouter context={{} location={req.path}>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop:50}}>
          {routes}
        </div>
      </Fragment>
    </StaticRouter>
  );
  res.send(`
    珠峰SSR

    ${content}
  `);
}
```

## 9. redux计数器

src/client/index.js

```
import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {Provider} from 'react-redux';
import createStore from '../store';
ReactDOM.hydrate(
  <Provider store={getStore()}>
    <BrowserRouter>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop:50}}>
          {routes}
        </div>
      </Fragment>
    </BrowserRouter>
  </Provider>
, document.querySelector('#root'));
```

src/containers/Counter/index.js

```
import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../store/actions';
class Counter extends Component{
  render() {
    return (
      <div>
        <p>{this.props.number}</p>
        <button className="btn btn-primary" onClick={this.props.increment}>+button</button>
      </div>
    )
  }
}
export default connect(
  state => state,
  actions
)(Counter);
```

src/server/render.js

```
import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import createStore from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  const content=renderToString(
    <Provider store={getStore()}>
      <StaticRouter context={{}} location={req.path}>
        <Fragment>
          <Header/>
          <div className="container" style={{marginTop:50}}>
            {routes}
          </div>
        </Fragment>
      </StaticRouter>
    </Provider>
  );
  res.send(`
    珠峰SSR
    ${content}
  `);
}
```

src/store/index.js

```
import reducer from './reducer';
import {createStore, applyMiddleware} from 'redux';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
function getStore() {
  return createStore(reducer, applyMiddleware(thunk, logger));
}
export default getStore;
```

src/store/reducer.js

```
import * as types from './action-types';
let initState = {
  number: 0
};
export default function (state = initState, action) {
  switch (action.type) {
    case types.INCREMENT:
      return {
        number: state.number + 1
      };
    default:
      return state;
  }
}
```

src/store/action-types.js

```
export const INCREMENT='INCREMENT';
```

src/store/actions.js

```
import * as types from '../action-types';
export default {
  increment() {
    return {type:types.INCREMENT}
  }
}
```

## 10. 合并reducers

src/containers/Counter/index.js

```
import React,{Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../../store/actions';
class Counter extends Component{
  render() {
    return (
      <div>
        <p>{this.props.number}</p>
        <button className="btn btn-primary" onClick={this.props.increment}>+button</button>
      </div>
    )
  }
}
export default connect(
  state => state.counter,
  actions
)(Counter);
```

src/store/index.js

```
import reducers from '../reducers';
import {createStore,applyMiddleware} from 'redux';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
function getStore() {
  return createStore(reducers,applyMiddleware(thunk,logger));
}
export default getStore;
```

src/store/reducers/index.js

```
import {combineReducers} from 'redux';
import home from './home';
import counter from './counter';
let reducers=combineReducers({
  home,
  counter
});
export default reducers;
```

src/store/reducers/home.js

```
import * as types from '../action-types';
let initState = {
  list:[]
};
export default function (state = initState, action) {
  switch (action.type) {
    default:
      return state;
  }
}
```

src/store/reducers/counter.js

```
import * as types from '../action-types';
let initState = {
  number: 0
};
export default function (state = initState, action) {
  switch (action.type) {
    case types.INCREMENT:
      return {
        number: state.number + 1
      };
    default:
      return state;
  }
}
```

## 11. 客户端加载数据

src/client/index.js

```
import React,{Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {Provider} from 'react-redux';
import getStore from '../store';
ReactDOM.hydrate(
  <Provider store={getStore()}>
    <BrowserRouter>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop:70}}>
          {routes}
        </div>
      </Fragment>
    </BrowserRouter>
  </Provider>
,document.querySelector('#root'));
```

src/containers/Counter/index.js

```
import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../../store/actions/counter';
class Counter extends Component{
  render() {
    return (
      <div>
        <p>{this.props.number}</p>
        <button className="btn btn-primary" onClick={this.props.increment}>+button</button>
      </div>
    )
  }
}
export default connect(
  state => state.counter,
  actions
)(Counter);
```

src/containers/Home/index.js

```
import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../../store/actions/home';
class Home extends Component{
  componentDidMount() {
    this.props.getHomeList();
  }
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <ul className="list-group">
            {
              this.props.list.map(item => (
                <li className="list-group-item" key={item.id}>{item.name}</li>
              ))
            }
          </ul>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.home,
  actions
)(Home);
```

src/containers/Home/index.js

```
import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../../store/actions/home';
class Home extends Component{
  componentDidMount() {
    this.props.getHomeList();
  }
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <ul className="list-group">
            {
              this.props.list.map(item => (
                <li className="list-group-item" key={item.id}>{item.name}</li>
              ))
            }
          </ul>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.home,
  actions
)(Home);
```

src/server/render.js



```
import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import createStore from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  const content=renderToString(
    <Provider store={getStore()}>
      <StaticRouter context={{}} location={req.path}>
        <Fragment>
          <Header/>
          <div className="container" style={{marginTop:70}}>
            {routes}
          </div>
        </Fragment>
      </StaticRouter>
    </Provider>
  );
  res.send(`
    珠峰SSR
    ${content}
  `);
}
```

src/store/action-types.js

```
export const INCREMENT='INCREMENT';

export const SET_HOME_LIST='SET_HOME_LIST';
```

src/store/reducers/home.js

```
import * as types from '../action-types';
let initState = {
  list:[]
};
export default function (state = initState, action) {
  switch (action.type) {
    case types.SET_HOME_LIST:
      return {...state,list:action.payload};
    default:
      return state;
  }
}
```

src/store/actions/counter.js

```
import * as types from '../action-types';
export default {
  increment() {
    return {type:types.INCREMENT}
  }
}
```

src/store/reducers/home.js

```
import * as types from '../action-types';
import axios from 'axios';
export default {
  getHomeList() {
    return function (dispatch,getState) {
      axios.get('http://localhost:4000/api/users').then(result => {
        let list=result.data;
        dispatch({
          type: types.SET_HOME_LIST,
          payload:list
        });
      });
    }
  }
}
```

api/server.js

```
let express=require('express');
let cors=require('cors');
let app=express();
var corsOptions = {
  origin: 'http://localhost:9090',
  optionsSuccessStatus: 200
}
app.use(cors(corsOptions));
let users=[{id:1,name:'zfpx1'},{id:2,name:'zfpx2'}];
app.get('/api/users',function (req,res) {
  res.json(users);
});
app.listen(4000);
```

## 12. 服务器端路由

```

import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {Provider} from 'react-redux';
import {Route} from 'react-router-dom';
import {getStore} from '../store';
ReactDOM.hydrate(
  <Provider store={getStore()}>
    <BrowserRouter>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {routes.map(route => (
              <Route {...route}/>
            ))}
          </Fragment>
        </div>
      </Fragment>
    </BrowserRouter>
  </Provider>
, document.querySelector('#root'));

```

src/containers/Home/index.js

```

import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../store/actions/home';
class Home extends Component {
  static loadData() => {
    console.log('加载数据');
  }

  componentDidMount() {
    this.props.getHomeList();
  }

  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <ul className="list-group">
            {
              this.props.list.map(item => (
                <li className="list-group-item" key={item.id}>{item.name}</li>
              ))
            }
          </ul>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.home,
  actions
)(Home);

```

src/routes.js

```

import React, {Fragment} from 'react';
import {Route} from 'react-router-dom';
import Home from '../containers/Home';
import Counter from '../containers/Counter';
export default [
  {
    path: '/',
    component: Home,
    exact: true,
    key: 'home',
    loadData: Home.loadData
  },
  {
    path: '/counter',
    component: Counter,
    key: 'login',
    exact: true
  }
]

```

src/server/render.js

```
import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {Route, matchPath} from 'react-router-dom';
import {getStore} from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  let store = getStore();
  let matchedRoutes = routes.filter(route => {
    return matchPath(req.path, route);
  });
  console.log(matchedRoutes);
  const content = renderToString(
    <Provider store={store}>
      <StaticRouter context={{}} location={req.path}>
        <Fragment>
          <Header/>
          <div className="container" style={{marginTop: 70}}>
            {routes.map(route => (
              <Route {...route}/>
            ))}
          </div>
        </Fragment>
      </StaticRouter>
    </Provider>
  );
  res.send(`
    珠峰SSR

    ${content}
  `);
}
```

### 13. 多级路由

src/client/index.js

```
import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {Provider} from 'react-redux';
import {Route, matchPath} from 'react-router-dom';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
ReactDOM.hydrate(
  <Provider store={getStore()}>
    <BrowserRouter>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {renderRoutes(routes)}
          </Fragment>
        </div>
      </Fragment>
    </BrowserRouter>
  </Provider>,
  document.querySelector('#root')
);
```

src/components/Header/index.js

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
export default class Home extends Component {
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">HomeLink</li>
              <li><Link to="/user/list">用户列表Link</li>
              <li><Link to="/counter">CounterLink</li>
            </ul>
          </div>
        </div>
      </nav>
    );
  }
}
```

src/routes.js

```

import React, {Fragment} from 'react';
import Home from '../containers/Home';
import User from '../containers/User';
import UserList from '../containers/User/components/UserList';
import Counter from '../containers/Counter';
export default [
  {
    path: '/',
    component: Home,
    exact: true,
    key: '/home',
    loadData: Home.loadData
  },
  {
    path: '/user',
    component: User,
    key: '/user',
    routes: [
      {
        path: '/user/list',
        component: UserList,
        key: '/user/list'
      }
    ]
  },
  {
    path: '/counter',
    component: Counter,
    key: 'login',
    exact: true
  }
]

```

src/server/render.js

```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {Route, matchPath} from 'react-router-dom';
import {matchRoutes, renderRoutes} from 'react-router-config';
import getStore from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  let store = getStore();

  let matchedRoutes = matchRoutes(routes, req.path);
  console.log(matchedRoutes);
  const content = renderToString(
    <Provider store={store}>
      <StaticRouter context={{}} location={req.path}>
        <Fragment>
          <Header/>
          <div className="container" style={{marginTop: 70}}>
            {renderRoutes(routes)}
          </div>
        </Fragment>
      </StaticRouter>
    </Provider>
  );
  res.send(`
    珠峰SSR
    ${content}
  `);
}

```

src/server/render.js

```

import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import {matchRoutes, renderRoutes} from 'react-router-config';
export default class User extends Component {
  render() {
    console.log(this.props.children);
    return (
      <div className="row">
        <div className="col-md-3">
          <ul className="list-group">
            <li className="list-group-item"><Link to="/user/list">用户列表</Link></li>
            <li className="list-group-item"><Link to="/user/add">添加用户</Link></li>
          </ul>
        </div>
        <div className="col-md-9">
          {renderRoutes(this.props.route.routes)}
        </div>
      </div>
    );
  }
}

```

src/containers/User/components/UserList.js

```

import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../store/actions/home';

class UserList extends Component {
  static loadData={() => {
    console.log('加载数据');
  }}

  componentDidMount() {
    this.props.getHomeList();
  }

  render() {
    return (
      <ul className="list-group">
        {
          this.props.list.map(item => (
            <li className="list-group-item" key={item.id}>{item.name}</li>
          ))
        }
      </ul>
    )
  }
}

export default connect(
  state => state.home,
  actions
)(UserList);

```

## 14. 后台获取数据

src/client/index.js

```

import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import Header from '../../components/Header';
import routes from '../../routes';
import {Provider} from 'react-redux';
import {renderRoutes} from 'react-router-config';
import {getClientStore} from '../../store';

ReactDOM.hydrate(
  <Provider store={getClientStore()}>
    <BrowserRouter>
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {renderRoutes(routes)}
          </Fragment>
        </div>
      </Fragment>
    </BrowserRouter>
  </Provider>,
  document.querySelector('#root')
);

```

src/containers/Home/index.js

```

import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../../store/actions/home';

class Home extends Component {
  static loadData=(store) => {
    return store.dispatch(actions.getHomeList());
  }

  componentDidMount() {
    if(this.props.list.length==0)
      this.props.getHomeList();
  }

  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <ul className="list-group">
            {
              this.props.list.map(item => (
                <li className="list-group-item" key={item.id}>{item.name}</li>
              ))
            }
          </ul>
        </div>
      </div>
    )
  }
}

export default connect(
  state => state.home,
  actions
)(Home);

```

src/server/render.js

```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {Route, matchPath} from 'react-router-dom';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  let store=getStore();

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={{}} location={req.path}>
          <Fragment>
            <Header/>
            <div className="container" style={{marginTop:70}}>
              {renderRoutes(routes)}
            </div>
          </Fragment>
        </StaticRouter>
      </Provider>
    );
    res.send(`
      珠峰SSR

      ${content}

      window.context = {
        state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span>.<span>stringify</span>(store.getState())}</span>
      }
    `);
  });
}

```

src/store/actions/home.js

```

import * as types from '../action-types';
import axios from 'axios';
export default {
  getHomeList() {
    return function (dispatch, getState) {
      return axios.get('http://localhost:4000/api/users').then(result => {
        let list=result.data;
        dispatch({
          type: types.SET_HOME_LIST,
          payload:list
        });
      });
    }
  }
}

```

src/store/index.js

```

import reducers from './reducers';
import {createStore, applyMiddleware} from 'redux';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
export function getStore() {
  return createStore(reducers, applyMiddleware(thunk, logger));
}
export function getClientStore() {
  let initState=window.context.state;
  return createStore(reducers, initState, applyMiddleware(thunk, logger));
}

```

## 15. Node代理服务器

src/server/index.js

```

import express from 'express';
import proxy from 'express-http-proxy';
import render from './render';
let app=express();
app.use(express.static('public'));
app.use('/api', proxy('http://127.0.0.1:4000', {
  proxyReqPathResolver: function (req) {
    return '/api/${req.url}';
  }
}));

app.get('*', (req, res) => {
  render(req, res);
});
app.listen(9090);

```

src/server/render.js

```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
export default function (req, res) {
  let store=getStore();

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={} location={req.path}>
          <Fragment>
            <Header/>
            <div className="container" style={{marginTop:70}}>
              {renderRoutes(routes)}
            </div>
          </Fragment>
        </StaticRouter>
      </Provider>
    );
    res.send(`
      珠峰SSR

      ${content}

      window.context = {
        state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span>.<span class="hljs-built_in">stringify</span>(store.getState())}</span>
      }
    `);
  });
}

```

src/store/actions/home.js

```

import * as types from '../action-types';
import axios from 'axios';
export default {
  getHomeList() {
    return function (dispatch, getState, request) {
      return request.get('/api/users').then(result => {
        let list=result.data;
        dispatch({
          type: types.SET_HOME_LIST,
          payload:list
        });
      });
    }
  }
}

```

src/store/index.js

```

import reducers from './reducers';
import {createStore, applyMiddleware} from 'redux';
import clientRequest from '../client/request';
import serverRequest from '../server/request';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
export function getStore() {
  return createStore(reducers, applyMiddleware(thunk.withExtraArgument(serverRequest), logger));
}
export function getClientStore() {
  let initState=window.context.state;
  return createStore(reducers, initState, applyMiddleware(thunk.withExtraArgument(clientRequest), logger));
}

```

src/client/request.js

```

import axios from 'axios';
export default axios.create({
  baseURL: '/'
});

```

src/server/request.js

```

import axios from 'axios';
export default axios.create({
  baseURL: 'http://localhost:4000/'
});

```

## 16. 抽取App.js

src/client/index.js

```

import React, {Fragment} from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import {Provider} from 'react-redux';
import routes from '../routes';
import {renderRoutes} from 'react-router-config';
import {getClientStore} from '../store';
ReactDOM.hydrate(
  <Provider store={getClientStore()}>>
    <BrowserRouter>
      {renderRoutes(routes)}
    </BrowserRouter>
  </Provider>
, document.querySelector('#root'));

```

src/containers/User/index.js

```

import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import {matchRoutes, renderRoutes} from 'react-router-config';
export default class User extends Component {
  render() {
    return (
      <div className="row">
        <div className="col-md-3">
          <ul className="list-group">
            <li className="list-group-item"><Link to="/user/list">用户列表Link</li>
            <li className="list-group-item"><Link to="/user/add">添加用户Link</li>
          </ul>
        </div>
        <div className="col-md-9">
          {renderRoutes(this.props.route.routes)}
        </div>
      </div>
    )
  }
}

```

src/routes.js

```

import React, {Fragment} from 'react';
import Home from './containers/Home';
import User from './containers/User';
import UserList from './containers/User/components/UserList';
import Counter from './containers/Counter';
import App from './containers/App';
export default [
  {
    path: '/',
    component: App,
    routes: [
      {
        path: '/',
        component: Home,
        exact: true,
        key: '/home',
        loadData: Home.loadData
      },
      {
        path: '/user',
        component: User,
        key: '/user',
        routes: [
          {
            path: '/user/list',
            component: UserList,
            key: '/user/list'
          }
        ]
      },
      {
        path: '/counter',
        component: Counter,
        key: 'login',
        exact: true
      }
    ]
  }
]

```

src/server/index.js

```

import express from 'express';
import proxy from 'express-http-proxy';
import render from './render';
let app=express();
app.use(express.static('public'));
app.use('/api', proxy('http://localhost:4000', {
  proxyReqPathResolver: function (req) {
    return `/api${req.url}`;
  }
}));
app.get('*', (req, res) => {
  render(req, res);
});
app.listen(9090);

```

src/server/render.js



```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
import App from '../containers/App';
export default function (req, res) {
  let store=getStore();

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={({}) location={req.path}>
          {renderRoutes(routes)}
        </StaticRouter>
      </Provider>
    );
    res.send(`
      珠峰SSR

      ${content}

      window.context = {
        state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span></span>.stringify(store.getState())}</span>
      }
    `);
  });
}

```

src/containers/App.js

```

import React, {Component, Fragment} from 'react';
import {renderRoutes} from 'react-router-config';
import Header from '../components/Header';
export default class App extends Component{
  render() {
    return (
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {renderRoutes(this.props.route.routes)}
          </Fragment>
        </div>
      </Fragment>
    )
  }
}

```

## 17. 实现头部导航

src/client/index.js

```

import React from 'react';
import ReactDOM from 'react-dom';
import {BrowserRouter} from 'react-router-dom';
import {Provider} from 'react-redux';
import routes from '../routes';
import {renderRoutes} from 'react-router-config';
import {getClientStore} from '../store';
ReactDOM.hydrate(
  <Provider store={getClientStore()}>
    <BrowserRouter>
      {renderRoutes(routes)}
    </BrowserRouter>
  </Provider>
, document.querySelector('#root'));

```

src/components/Header/index.js

```

import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import {connect} from 'react-redux';
class Header extends Component{
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">首页Link</li>
              <li><Link to="/user/list">用户列表Link</li>
              {
                !this.props.user&&<li><Link to="/login">登录Link</li>
              }
              {
                this.props.user&&<Fragment><li><Link to="/logout">退出Link</li><li><Link to="/profile">个人中心Link</li></Fragment>
              }
            </ul>
          </div>
        </div>
      </nav>
    )
  }
}
export default connect(
  state=>state.session
)(Header);

```

src/store/action-types.js

```

export const INCREMENT='INCREMENT';

export const SET_HOME_LIST='SET_HOME_LIST';

export const LOGIN='LOGIN';

export const LOGOUT='LOGOUT';

```

src/store/reducers/index.js

```

import {combineReducers} from 'redux';
import home from './home';
import counter from './counter';
import session from './session';
let reducers=combineReducers({
  home,
  counter,
  session
});
export default reducers;

```

src/store/reducers/session.js

```

import * as types from '../action-types';
let initState = {
  user:null
};
export default function (state = initState, action) {
  switch (action.type) {
    default:
      return state;
  }
}

```

## 18. 实现登录功能

```

let express=require('express');
let cors=require('cors');
let morgan=require('morgan');
let session=require('express-session');
let bodyParser=require('body-parser');
let app=express();
app.use(morgan('tiny'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(session({
  resave: true,
  saveUninitialized: true,
  secret:'zfxp'
}));
var corsOptions = {
  origin: '*',
  optionsSuccessStatus: 200
}
app.use(cors(corsOptions));
let users=[{id:1,name:'zfxp1'},{id:2,name:'zfxp2'}];
app.get('/api/users',function (req,res) {
  res.json(users);
});
app.post('/api/login',function (req,res) {
  let user=req.body;
  req.session.user=user;
  res.json({
    code: 0,
    user,
    success:'登录成功'
  });
});
app.get('/api/logout',function (req,res) {
  req.session.user=null;
  res.json({
    code: 0,
    success:'退出成功'
  });
});
app.get('/api/user',function (req,res) {
  if (req.session.user) {
    res.json({
      code: 0,
      user,
      success:'获取用户信息成功'
    });
  } else {
    res.json({
      code: 1,
      error:'此用户未登录'
    });
  }
});
app.listen(4000);

```

src/components/Header/index.js

```

import React,{Component,Fragment} from 'react';
import {Link} from 'react-router-dom';
import {connect} from 'react-redux';
class Header extends Component{
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">首页Link</li>
              <li><Link to="/user/list">用户列表Link</li>
              {
                !this.props.user&&<li><Link to="/login">登录Link</li>
              }
              {
                this.props.user&&<Fragment><li><Link to="/logout">退出Link</li><li><Link to="/profile">个人中心Link</li></Fragment>
              }
            </ul>
            {
              this.props.user&&(
                <ul class="nav navbar-nav navbar-right">
                  <li><a href="#">欢迎 {this.props.user.username}</li>
                </ul>
              )
            }
          </div>
        </div>
      </nav>
    )
  }
}
export default connect(
  state=>state.session
)(Header);

```

src/containers/App.js

```

import React, {Component, Fragment} from 'react';
import {renderRoutes} from 'react-router-config';
import Header from '../components/Header';
export default class App extends Component{
  static loadData=(store) => {
    store.dispatch();
  }
  render() {
    return (
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {renderRoutes(this.props.route.routes)}
          </Fragment>
        </div>
      </Fragment>
    )
  }
}

```

src/routes.js

```

import React, {Fragment} from 'react';
import Home from './containers/Home';
import User from './containers/User';
import UserList from './containers/User/components/UserList';
import Counter from './containers/Counter';
import Login from './containers/Login';
import Logout from './containers/Logout';
import Profile from './containers/Profile';
import App from './containers/App';
export default [
  {
    path: '/',
    component: App,
    routes: [
      {
        path: '/',
        component: Home,
        exact: true,
        key: '/home',
        loadData: Home.loadData
      },
      {
        path: '/user',
        component: User,
        key: '/user',
        routes: [
          {
            path: '/user/list',
            component: UserList,
            key: '/user/list'
          }
        ]
      },
      {
        path: '/counter',
        component: Counter,
        key: '/counter',
        exact: true
      },
      {
        path: '/login',
        component: Login,
        key: '/login',
        exact: true
      },
      {
        path: '/logout',
        component: Logout,
        key: '/logout',
        exact: true
      },
      {
        path: '/profile',
        component: Profile,
        key: '/profile',
        exact: true
      }
    ]
  }
]

```

src/server/index.js

```
import express from 'express';
import proxy from 'express-http-proxy';
import render from './render';
let app=express();
app.use(express.static('public'));

app.use('/api',proxy('http://localhost:4000',{

  proxyReqPathResolver: function (req) {
    return `/api${req.url}`;
  }
}));

app.get('*', (req,res) => {
  render(req,res);
});
app.listen(9090);
```

src/store/action-types.js

```
export const INCREMENT='INCREMENT';

export const SET_HOME_LIST='SET_HOME_LIST';

export const LOGIN='LOGIN';

export const LOGOUT='LOGOUT';

export const SET_SESSION='SET_SESSION';
```

src/store/reducers/session.js

```
import * as types from '../action-types';
let initState = {
  user: null,
  success: null,
  error:null
};
export default function (state = initState, action) {
  switch (action.type) {
    case types.SET_SESSION:
      return {...action.payload};
    default:
      return state;
  }
}
```

src/containers/Login/index.js

```
import React,{Component} from 'react';
import actions from '../../../store/actions/session';
import {connect} from 'react-redux';
class Login extends Component{
  state={
    username:''
  }
  handleChange=(event) => {
    this.setState({username:event.target.value});
  }
  handleSubmit=(event) => {
    event.preventDefault();
    this.props.login(this.state);
  }
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <form onSubmit={this.handleSubmit}>
            <div className="form-group">
              <label htmlFor="username">用户名label</label>
              <input value={this.state.username} onChange={this.handleChange} type="text" className="form-control"/>
            </div>
            <div className="form-group">
              <input type="submit" className="btn btn-primary"/>
            </div>
          </form>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.session,
  actions
)(Login)
```

src/containers/Logout/index.js

```
import React,{Component} from 'react';
export default class Profile extends Component{
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <button className="btn btn-primary">退出button</button>
        </div>
      </div>
    )
  }
}
```

src/containers/Profile/index.js

```
import React, {Component} from 'react';
export default class Profile extends Component {
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          个人中心
        </div>
      </div>
    )
  }
}
```

src/store/actions/session.js

```
import * as types from '../action-types';
export default {
  login(user) {
    return function (dispatch, getState, request) {
      return request.post('/api/login', user).then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload: result.data
        });
      });
    }
  }
}
```

## 19. 退出功能

src/containers/Logout/index.js

```
import React, {Component} from 'react';
import actions from '../../../store/actions/session';
import {connect} from 'react-redux';
class Profile extends Component {
  handleLogout=() => {
    this.props.logout();
  }
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <button className="btn btn-primary" onClick={this.handleLogout}>退出button</button>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.session,
  actions
)(Profile)
```

src/store/actions/session.js

```
import * as types from '../action-types';
export default {
  login(user) {
    return function (dispatch, getState, request) {
      return request.post('/api/login', user).then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload: result.data
        });
      });
    }
  },
  logout() {
    return function (dispatch, getState, request) {
      return request.get('/api/logout').then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload: result.data
        });
      });
    }
  }
}
```

## 20. 加载用户信息

```

let express=require('express');
let cors=require('cors');
let morgan=require('morgan');
let session=require('express-session');
let bodyParser=require('body-parser');
let app=express();
app.use(morgan('tiny'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: true}));
app.use(session({
  resave: true,
  saveUninitialized: true,
  secret:'zfpx'
}));
var corsOptions = {
  origin: '**',
  optionsSuccessStatus: 200
}
app.use(cors(corsOptions));
let users=[{id:1,name:'zfpx1'},{id:2,name:'zfpx2'}];
app.get('/api/users',function (req,res) {
  res.json(users);
});
app.post('/api/login',function (req,res) {
  let user=req.body;
  req.session.user=user;
  res.json({
    code: 0,
    data: {
      user,
      success:'登录成功'
    }
  });
});
app.get('/api/logout',function (req,res) {
  req.session.user=null;
  res.json({
    code: 0,
    data: {
      success:'退出成功'
    }
  });
});
app.get('/api/user',function (req,res) {
  if (req.session.user) {
    res.json({
      code: 0,
      data: {
        user:req.session.user,
        success:'获取用户信息成功'
      }
    });
  } else {
    res.json({
      code: 1,
      data: {
        error:'此用户未登录'
      }
    });
  }
});
app.listen(4000);

```

src/containers/App.js

```

import React,{Component,Fragment} from 'react';
import {renderRoutes} from 'react-router-config';
import Header from '../components/Header';
import actions from '../store/actions/session';
export default class App extends Component{
  static loadData=(store) => {
    return store.dispatch(actions.getUser());
  }
  render() {
    return (
      <Fragment>
        <Header/>
        <div className="container" style={{marginTop: 70}}>
          <Fragment>
            {renderRoutes(this.props.route.routes)}
          </Fragment>
        </div>
      </Fragment>
    )
  }
}

```

src/containers/Profile/index.js

```

import React, {Component} from 'react';
import actions from '../../store/actions/session';
import {Redirect} from 'react-router-dom';
import {connect} from 'react-redux';
class Profile extends Component{
  render() {
    return this.props.user? (
      <div className="row">
        <div className="col-md-12">
          {this.props.user.username}
        </div>
      </div>
    ):<Redirect to="/login"/>;
  }
}
export default connect(
  state => state.session,
  actions
)(Profile)

```

src/routes.js

```

import React, {Fragment} from 'react';
import Home from './containers/Home';
import User from './containers/User';
import UserList from './containers/User/components/UserList';
import Counter from './containers/Counter';
import Login from './containers/Login';
import Logout from './containers/Logout';
import Profile from './containers/Profile';
import App from './containers/App';
export default [
  {
    path: '/',
    component: App,
    loadData: App.loadData,
    routes: [
      {
        path: '/',
        component: Home,
        exact: true,
        key: '/home',
        loadData: Home.loadData
      },
      {
        path: '/user',
        component: User,
        key: '/user',
        routes: [
          {
            path: '/user/list',
            component: UserList,
            key: '/user/list'
          }
        ]
      }
    ],
  },
  {
    path: '/counter',
    component: Counter,
    key: '/counter',
    exact: true
  },
  {
    path: '/login',
    component: Login,
    key: '/login',
    exact: true
  },
  {
    path: '/logout',
    component: Logout,
    key: '/logout',
    exact: true
  },
  {
    path: '/profile',
    component: Profile,
    key: '/profile',
    exact: true
  }
]
]

```

src/server/render.js



```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
import App from '../containers/App';
export default function (req, res) {
  let store=getStore(req);

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={{} location={req.path}>
          {renderRoutes(routes)}
        </StaticRouter>
      </Provider>
    );
    res.send(`

    珠峰SSR

    ${content}

    window.context = {
      state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span></span>.stringify(store.getState())}</span>
    }

  `);
  });
}

```

src/server/request.js

```

import axios from 'axios';
export default (req)=>axios.create({
  baseURL: 'http://localhost:4000/',
  headers: {
    cookie:req.get('cookie') || ''
  }
});

```

src/store/actions/session.js

```

import * as types from '../action-types';
export default {
  login(user) {
    return function (dispatch, getState, request) {
      return request.post('/api/login', user).then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload:result.data.data
        });
      });
    }
  },
  logout() {
    return function (dispatch, getState, request) {
      return request.get('/api/logout').then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload:result.data.data
        });
      });
    }
  },
  getUser() {
    return function (dispatch, getState, request) {
      return request.get('/api/user').then(result => {
        dispatch({
          type: types.SET_SESSION,
          payload:result.data.data
        });
      });
    }
  }
}

```

src/store/index.js

```

import reducers from './reducers';
import {createStore,applyMiddleware} from 'redux';
import clientRequest from '../client/request';
import getServerRequest from '../server/request';
import thunk from 'redux-thunk';
import logger from 'redux-logger';
export function getStore(req) {
    return createStore(reducers,applyMiddleware(thunk.withExtraArgument(getServerRequest(req)),logger));
}
export function getClientStore() {
    let initState=window.context.state;
    return createStore(reducers,initState,applyMiddleware(thunk.withExtraArgument(clientRequest),logger));
}

```

## 21. 404

```

import Home from './containers/Home';
import User from './containers/User';
import UserList from './containers/User/components/UserList';
import Counter from './containers/Counter';
import Login from './containers/Login';
import Logout from './containers/Logout';
import Profile from './containers/Profile';
import NotFound from './containers/NotFound';
import App from './containers/App';
export default [
    {
        path: '/',
        component: App,
        loadData:App.loadData,
        routes: [
            {
                path: '/',
                component: Home,
                exact: true,
                key:'/home',
                loadData:Home.loadData
            },
            {
                path: '/user',
                component: User,
                key: '/user',
                routes: [
                    {
                        path: '/user/list',
                        component: UserList,
                        key:'/user/list'
                    }
                ]
            },
        ],
    },
    {
        path: '/counter',
        component: Counter,
        key:'counter',
        exact: true
    },
    {
        path: '/login',
        component: Login,
        key:'/login',
        exact: true
    },
    {
        path: '/logout',
        component: Logout,
        key:'/logout',
        exact: true
    },
    {
        path: '/profile',
        component: Profile,
        key:'/profile',
        exact: true
    },
    {
        component: NotFound
    }
]
]

```

src/server/render.js

```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
import App from '../containers/App';
export default function (req, res) {
  let store=getStore(req);

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    let context={};
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={context} location={req.path}>
          {renderRoutes(routes)}
        </StaticRouter>
      </Provider>
    );
    if (context.notFound) {
      res.status(404);
    }
    res.send(`
      珠峰SSR

      ${content}

      window.context = {
        state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span>.stringify(store.getState())}</span>
      }
    `);
  });
}

```

src/containers/NotFound/index.js

```

import React, {Component} from 'react';
export default class NotFound extends Component{
  componentWillMount() {
    if (this.props.staticContext) {
      this.props.staticContext.notFound=true;
    }
  }
  render() {
    return (
      <div>404div</div>
    )
  }
}

```

## 22. 301

src/server/render.js

```

import React, {Component, Fragment} from 'react';
import {StaticRouter} from 'react-router-dom';
import Header from '../components/Header';
import routes from '../routes';
import {renderToString} from 'react-dom/server';
import {matchRoutes, renderRoutes} from 'react-router-config';
import {getStore} from '../store';
import {Provider} from 'react-redux';
import App from '../containers/App';
export default function (req, res) {
  let store=getStore(req);

  let matchedRoutes=matchRoutes(routes, req.path);
  let promises=[];
  matchedRoutes.forEach(item => {
    if (item.route.loadData)
      promises.push(item.route.loadData(store));
  });
  Promise.all(promises).then(result => {
    let context={};
    const content=renderToString(
      <Provider store={store}>
        <StaticRouter context={context} location={req.path}>
          {renderRoutes(routes)}
        </StaticRouter>
      </Provider>
    );
    if (context.action == 'REPLACE') {
      return res.redirect(301, context.url);
    } else if (context.notFound) {
      res.status(404);
    }
    res.send(`
      珠峰SSR

      ${content}

      window.context = {
        state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span>.stringify(store.getState())}</span>
      }
    `);
  });
}

```

## 23. promise.all

```

let promises=[];
matchedRoutes.forEach(item => {
  if (item.route.loadData) {
    let promise=new Promise(function (resolve, reject) {
      return item.route.loadData(store).then(resolve, resolve);
    });
    promises.push(promise);
  }
});

```

## 24. 使用CSS

src/components/Header/index.js

```

import React, {Component, Fragment} from 'react';
import {Link} from 'react-router-dom';
import {connect} from 'react-redux';
import styles from './index.css';
class Header extends Component{
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">首页Link</li>
              <li><Link to="/user/list">用户列表Link</li>
              {
                !this.props.user&&<li><Link to="/login">登录Link</li>
              }
              {
                this.props.user&&<Fragment><li><Link to="/logout">退出Link</li><li><Link to="/profile">个人中心Link</li></Fragment>
              }
            </ul>
            {
              this.props.user&&(
                <ul className="nav navbar-nav navbar-right">
                  <li><a href="#">欢迎 <span className={styles.user}>{this.props.user.username}</span><a>li>
                </ul>
              )
            }
          </div>
        </div>
      </nav>
    )
  }
}
export default connect(
  state=>state.session
)(Header);

```

src/containers/App.js

```
import React, {Component, Fragment} from 'react';
import {renderRoutes} from 'react-router-config';
import Header from '../components/Header';
import actions from '../store/actions/session';
import styles from './App.css';
export default class App extends Component{
  static loadData=(store) => {
    return store.dispatch(actions.getUser());
  }
  render() {
    return (
      <Fragment>
        <Header/>
        <div className="container" className={styles.app}>
          <Fragment>
            {renderRoutes(this.props.route.routes)}
          </Fragment>
        </div>
      </Fragment>
    )
  }
}
```

webpack.client.js

```
module:{
  rules:[
    {
      test: /\.css$/,
      use: [
        'style-loader',
        {
          loader: 'css-loader',
          options: {
            modules: true,
            localIdentName: '[name]_[local]_[hash:base64:5]'
          }
        }
      ]
    }
  ]
}
```

webpack.server.js

```
module:{
  rules:[
    {
      test: /\.css$/,
      use: [
        'isomorphic-style-loader',
        {
          loader: 'css-loader',
          options: {
            modules: true,
            localIdentName: '[name]_[local]_[hash:base64:5]'
          }
        }
      ]
    }
  ]
}
```

src/components/Header/index.css

```
.user{
  color:red;
}
```

src/containers/App.css

```
.app{
  margin-top:70px;
}
```

## 25. CSS服务器端渲染

src/components/Header/index.js

```
+   componentWillMount() {
+     if (this.props.staticContext) {
+       this.props.staticContext.csses.push(styles._getCss());
+     }
+   }
```

src/containers/App.js

```
+   componentWillMount() {
+     if (this.props.staticContext) {
+       this.props.staticContext.csses.push(styles._getCss());
+     }
+   }
```

src/server/render.js

```

let context={csses:[]};
const content=renderToString(

    {renderRoutes(routes)}

);
let cssStr='';
+ if (context.csses.length>0) {
+   cssStr=context.csses.join('\r\n');
+ }
if (context.action == 'REPLACE') {
  return res.redirect(301,context.url);
} else if (context.notFound) {
  res.status(404);
}
res.send(`

+   ${cssStr}
  珠峰SSR

  ${content}

  window.context = {
    state:${JSON.stringify(store.getState())}
  }

`);

```

## 26. 优化CSS服务器端渲染

src/components/Header/index.js

```

import React,{Component,Fragment} from 'react';
import {Link} from 'react-router-dom';
import {connect} from 'react-redux';
import styles from './index.css';
import withStyles from '../../withStyles';
class Header extends Component{
  render() {
    return (
      <nav className="navbar navbar-inverse navbar-fixed-top">
        <div className="container">
          <div className="navbar-header">
            <a className="navbar-brand" href="#">珠峰SSRa</a>
          </div>
          <div id="navbar" className="collapse navbar-collapse">
            <ul className="nav navbar-nav">
              <li><Link to="/">首页Link</li>
              <li><Link to="/user/list">用户列表Link</li>
              {
                !this.props.user&&<li><Link to="/login">登录Link</li>
              }
              {
                this.props.user&&<Fragment><li><Link to="/logout">退出Link</li><li><Link to="/profile">个人中心Link</li></Fragment>
              }
            </ul>
            {
              this.props.user&&(
                <ul className="nav navbar-nav navbar-right">
                  <li><a href="#">欢迎 <span className={styles.user}>{this.props.user.username}</span></a></li>
                </ul>
              )
            }
          </div>
        </div>
      </nav>
    )
  }
}
export default connect(
  state=>state.session
)(withStyles(Header,styles));

```

src/containers/App.js

```

import React, {Component, Fragment} from 'react';
import {renderRoutes} from 'react-router-config';
import Header from '../components/Header';
import actions from '../store/actions/session';
import styles from './App.css';
import withStyles from '../withStyles';
class App extends Component{
  componentWillMount() {
    if (this.props.staticContext) {
      this.props.staticContext.csses.push(styles._getCss());
    }
  }
  render() {
    return (
      <Fragment>
        <Header staticContext={this.props.staticContext}/>
        <div className="container" className={styles.app}>
          <Fragment>
            {renderRoutes(this.props.route.routes)}
          </Fragment>
        </div>
      </Fragment>
    )
  }
}
let Proxy=withStyles(App,styles);
Proxy.loadData=(store) => {
  return store.dispatch(actions.getUser());
}
export default Proxy;

```

src/containers/Home/index.js

```

import React, {Component} from 'react';
import {connect} from 'react-redux';
import actions from '../store/actions/home';
import withStyles from '../withStyles';
class Home extends Component{
  static loadData=(store) => {
    return store.dispatch(actions.getHomeList());
  }
  componentWillMount() {
    if(this.props.list.length==0)
      this.props.getHomeList();
  }
  render() {
    return (
      <div className="row">
        <div className="col-md-12">
          <ul className="list-group">
            {
              this.props.list.map(item => (
                <li className="list-group-item" key={item.id}>{item.name}</li>
              ))
            }
          </ul>
        </div>
      </div>
    )
  }
}
export default connect(
  state => state.home,
  actions
)(Home);

```

src/withStyles.js

```

import React, {Component, Fragment} from 'react';

export default function withStyles(OriginalComponent, styles) {
  class ProxyComponent extends Component{
    componentWillMount() {
      if (this.props.staticContext) {
        this.props.staticContext.csses.push(styles._getCss());
      }
    }
    render() {
      return <OriginalComponent {...this.props}/>
    }
  }
  return ProxyComponent;
}

```

## 27. SEO优化

src/containers/Home/index.js

```

import React, {Component, Fragment} from 'react';
import {connect} from 'react-redux';
import {Helmet} from 'react-helmet';
import actions from '../../store/actions/home';

class Home extends Component {
  static loadData=(store) => {

    return store.dispatch(actions.getHomeList());
  }

  componentDidMount() {
    if(this.props.list.length==0)
      this.props.getHomeList();
  }

  render() {
    return (
      <Fragment>
        <Helmet>
          <title>首页标题title>
          <meta name="description" content="首页描述">meta</meta>
        </Helmet>
        <div className="row">
          <div className="col-md-12">
            <ul className="list-group">
              {
                this.props.list.map(item => (
                  <li className="list-group-item" key={item.id}>{item.name}</li>
                ))
              }
            </ul>
          </div>
        </div>
      </Fragment>
    )
  }
}

export default connect(
  state => state.home,
  actions
)(Home);

```

src/server/render.js

```

import {Helmet} from 'react-helmet';
let helmet=Helmet.renderStatic();

let cssStr='';
if (context.csses.length>0) {
  cssStr=context.csses.join('\r\n');
}
if (context.action == 'REPLACE') {
  return res.redirect(301,context.url);
} else if (context.notFound) {
  res.status(404);
}
res.send(`

  ${helmet.title.toString()}
  ${helmet.meta.toString()}

  <span class="hljs-subst">${cssStr}</span>
  珠峰SSR

  ${content}

  window.context = {
    state:<span class="hljs-subst">${<span class="hljs-built_in">JSON</span>.stringify(store.getState())}</span>
  }

`);

```