

link: null  
title: 珠峰架构师成长计划  
description: src/index.js  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=59 sentences=116, words=885

## 1.生成项目

```
create-react-app zhufeng_connected_router
cd zhufeng_connected_router
npm i redux react-redux react-router-dom connected-react-router -S
```

## 2.跑通项目

src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, Route, Link } from 'react-router-dom';
import Home from './components/Home';
import Counter from './components/Counter';
import history from './history';
import store from './store';
import { Provider } from 'react-redux';
ReactDOM.render(
  <>
    <Provider store={store}>
      <Router history={history}>
        <Route path="/" component={Home} />
        <Route path="/counter" component={Counter} />
      </Router>
    </Provider>
  </>,
  document.getElementById('root')
);
```

src/history.js

```
import { createBrowserHistory } from 'history'
let history = createBrowserHistory();
export default history;
```

src/store/index.js

```
import { applyMiddleware, createStore } from 'redux'
import reducers from './reducers';
const store = applyMiddleware()(createStore)(reducers);
export default store;
```

src/store/action-types.js

```
export const ADD = 'ADD';
export const MINUS = 'MINUS';
```

src/store/actionCreators/counter.js

```
import * as actionTypes from '../action-types';
const actionCreators = {
  add() {
    return { type: actionTypes.ADD };
  },
  minus() {
    return { type: actionTypes.MINUS };
  }
}
export default actionCreators
```

src/store/reducers/counter.js

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

src/store/reducers/index.js

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

src/components/Home.js

```
import React, { Component } from 'react';
export default class Home extends Component {
  render() {
    return (
      <div>
        <h1>Homeh1<
        <button onClick={() => this.props.history.go(-1)}>返回button>
      </div>
    )
  }
}
```

src/components/Counter.js

```
import React from 'react'
import { connect } from 'react-redux';
import actionCreators from '../store/actionCreators/counter';
class Counter extends React.Component {
  render() {
    return (
      <>
        <p>{this.props.number}</p>
        <button onClick={this.props.add}>+button</button>
        <button onClick={this.props.minus}>-button</button>
      </>
    )
  }
}
let mapStateToProps = (state) => state.counter;
export default connect(
  mapStateToProps,
  actionCreators
)(Counter);
```

### 3.同步路由状态到仓库

src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
+import { Route, Link } from 'react-router-dom';
import Home from './components/Home';
import Counter from './components/Counter';
import history from './history';
import store from './store';
import { Provider } from 'react-redux';
+import { ConnectedRouter } from './connected-react-router';
ReactDOM.render(
+
  <>
    Home
    Counter
  </>
+
, document.getElementById('root'));
```

src/store/reducers/index.js

```
import { combineReducers } from 'redux'
import counter from './counter';
+import history from '.././history';
+import { connectRouter } from '.././connected-react-router'
let reducers = {
+  router: connectRouter(history),
  counter
};
let rootReducer = combineReducers(reducers);
export default rootReducer;
```

src/connected-react-router/index.js

```
export { default as ConnectedRouter } from './ConnectedRouter';
export { default as connectRouter } from './connectRouter';
```

src/connected-react-router/actions.js

```
export const LOCATION_CHANGE = '@@router/LOCATION_CHANGE';
export const onLocationChange = (location, action) => ({
  type: LOCATION_CHANGE,
  payload: {
    action,
    location
  }
})
```

src/connected-react-router/ConnectedRouter.js

```
import React, { Component } from 'react'
import { ReactReduxContext } from 'react-redux';
import { onLocationChange } from './actions';
import { Router } from 'react-router-dom';

export default class ConnectedRouter extends Component {
  static contextType = ReactReduxContext;
  constructor(props, context) {
    super(props);

    this.unlisten = this.props.history.listen((location, action) => {
      context.store.dispatch(onLocationChange(location, action));
    });
  }
  componentWillUnmount() {
    this.unlisten();
  }
  render() {
    return (
      <Router history={this.props.history}>
        {this.props.children}
      </Router>
    )
  }
}
```

src\connected-react-router\connectRouter.js

```
import { LOCATION_CHANGE } from './actions';
function connectRouter(history) {
  const initialState = { action: history.action, location: history.location };
  return function reducer(state = initialState, { type, payload }) {
    if (type === LOCATION_CHANGE) {
      return { ...state, action: payload.action, location: payload.location };
    }
    return state;
  }
}

export default connectRouter;
```

#### 4. 派发动作修改路径

src\store\actionCreators\counter.js

```
import * as actionTypes from '../action-types';
+import { push } from '../../connected-react-router';
const actionCreators = {
  add() {
    return { type: actionTypes.ADD }
  },
  minus() {
    return { type: actionTypes.MINUS }
  },
+  goto(path) {
+    return push(path);
+  }
}
export default actionCreators
```

src\components\Counter.js

```
import React from 'react'
import { connect } from 'react-redux';
import actionCreators from '../store/actionCreators/counter';
class Counter extends React.Component {
  render() {
    return (
      <>
        {this.props.number}
        +
        -
+      {this.props.goto('/')}>跳转到/home里
      </>
    )
  }
}
let mapStateToProps = (state) => state.counter;
export default connect(
  mapStateToProps,
  actionCreators
)(Counter);
```

src\store\index.js

```
import { applyMiddleware, createStore } from 'redux'
import reducers from './reducers';
+import { routerMiddleware } from '../connected-react-router'
+import history from '../history';
+const store = applyMiddleware(routerMiddleware(history))(createStore)(reducers);
export default store;
```

src\connected-react-router\index.js

```
export { default as ConnectedRouter } from './ConnectedRouter';
export { default as connectRouter } from './connectRouter';
+export { push } from './actions';
+export { default as routerMiddleware } from './routerMiddleware';
```

src\connected-react-router\actions.js

```

export const LOCATION_CHANGE = '@@router/LOCATION_CHANGE';
export const onLocationChange = (location, action) => ({
  type: LOCATION_CHANGE,
  payload: {
    action,
    location
  }
})

+export const CALL_HISTORY_METHOD = '@@router/CALL_HISTORY_METHOD';
+export const push = (path) => ({
+  type: CALL_HISTORY_METHOD,
+  payload: {
+    method: 'push',
+    path
+  }
+})

```

src\connected-react-router\routerMiddleware.js

```

import { CALL_HISTORY_METHOD } from './actions';
function routerMiddleware(history) {
  return function ({ getState, dispatch }) {
    return function (next) {
      return function (action) {
        const { type, payload } = action
        if (type === CALL_HISTORY_METHOD) {
          history[payload.method](payload.path);
        } else {
          return next(action);
        }
      }
    }
  }
}

export default routerMiddleware;

```