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=47 sentences=91, words=671

1.生成项目

```
create-react-app zhufeng_redux_first_history
cd zhufeng_redux_first_history
npm install redux react-redux  react-router-dom redux-first-history --save
```

2.跑通项目

src\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Route, Routes, Link } from "react-router-dom";
import { HistoryRouter } from "./redux-first-history/rr6";
import { Provider } from 'react-redux';
import { store, history } from "./store";
import Home from './components/Home';
import Counter from './components/Counter';
ReactDOM.render(
  <Provider store={store}>
    <HistoryRouter history={history}>
      <ul>
        <li><Link to="/">HomeLink>li>
        <li><Link to="/counter">CounterLink>li>
      ul>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/counter" element={<Counter />} />
      Routes>
    HistoryRouter>
  Provider >,
  document.getElementById('root')
);
```

src\history.js

```
import { createBrowserHistory } from 'history';
import { createReduxHistoryContext } from "./redux-first-history";
const history = createBrowserHistory();
const { routerReducer, routerMiddleware, createReduxHistory } = createReduxHistoryContext({ history });
export {
    routerReducer,
    routerMiddleware,
    createReduxHistory
}
```

src\store\index.js

```
import { createStore, applyMiddleware } from 'redux';
import combinedReducer from './reducers';
import { routerMiddleware, createReduxHistory } from '../history';

export const store = applyMiddleware(routerMiddleware)(createStore)(combinedReducer);
window.store = store;
export const history = createReduxHistory(store);
```

src\store\action-types.js

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

src\store\reducers\counter.js

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

src\store\reducers\index.js

```
import { combineReducers } from 'redux';
import counter from './counter';
import { routerReducer } from '../../history';
const reducers = {
    counter,
    router: routerReducer
}
export default combineReducers(reducers);
```

src\components\Home.js

```
import React from 'react';
import { useDispatch } from 'react-redux';
import { push } from "../redux-first-history";
function Home() {

    const dispatch = useDispatch();
    const gotoCounter = () => {

        dispatch(push('/counter'));
    }
    return (
        <div>
            <p>Homep>
            <button onClick={gotoCounter}>跳转到/counterbutton>
        div>
    )
}
export default Home;
```

src\components\Counter.js

```
function Counter() {
    return (
        <div>
            <p>Counterp>
        div>
    )
}
export default Counter;
```

3.实现

src\redux-first-history\index.js

```
export { push } from './actions';
export { createReduxHistoryContext } from './create';
```

src\redux-first-history\actions.js

```

export const CALL_HISTORY_METHOD = '@@router/CALL_HISTORY_METHOD';

export const LOCATION_CHANGE = '@@router/LOCATION_CHANGE';

export function locationChangeAction(location, action) {
  return {
    type: LOCATION_CHANGE,
    payload: { action, location }
  }
}

function updateLocation(method) {
  return (...args) => {
    return {
      type: CALL_HISTORY_METHOD,
      payload: { method, args }
    }
  }
}

export const push = updateLocation('push');
```

src\redux-first-history\create.js

```
import { createRouterMiddleware } from './middleware';
import { push, locationChangeAction } from './actions';
import { createRouterReducer } from './reducer';

export function createReduxHistoryContext({ history }) {
  const routerMiddleware = createRouterMiddleware(history);
  const routerReducer = createRouterReducer(history);
  function createReduxHistory(store) {
    store.dispatch(locationChangeAction(history.location, history.action));

    history.listen(({ location, action }) => {
      store.dispatch(locationChangeAction(location, action));
    });
    return {
      createHref: history.createHref,
      push: (...args) => store.dispatch(push(...args)),
      listen: history.listen,
      get location() {
        return store.getState().router.location;
      },
      get action() {
        return store.getState().router.action;
      }
    };
  }
  return {
    routerMiddleware,
    createReduxHistory,
    routerReducer
  }
}
```

src\redux-first-history\middleware.js

```

import { CALL_HISTORY_METHOD } from './actions'

export function createRouterMiddleware(history) {
  return function () {
    return function (next) {
      return function (action) {
        if (action.type !== CALL_HISTORY_METHOD) {
          return next(action);
        }
        const { method, args } = action.payload;
        history[method](...args);
      }
    }
  }
}
```

src\redux-first-history\reducer.js

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

```

src\redux-first-history\rr6\index.js

```
import React from 'react';
import { Router } from 'react-router';

export function HistoryRouter({ history, children }) {
  const [state, setState] = React.useState({
    action: history.action,
    location: history.location
  });
  React.useLayoutEffect(() => {
    history.listen(setState);
  }, [history]);
  return (
    <Router
      location={state.location}
      action={state.action}
      navigator={history}
      navigationType={state.action}
    >
      {children}
    Router>
  )

}
```