React

## Redux

### Redux的基本使用(例子)

#### Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

import { createStore, applyMiddleware, compose } from 'redux'

import thunk from 'redux-thunk'

import { counter, addGun, removeGun, addGunAsync } from './index.redux'

// compose把多个中间件连接在一块

const store = createStore(counter, compose(

applyMiddleware(thunk),

window.devToolsExtension? window.devToolsExtension(): f=>f // 使用redux-devtools

))

function render() {

ReactDOM.render(<App store={ store } addGun={addGun} addGunAsync={addGunAsync} removeGun={removeGun} />, document.getElementById('root'));

}

render()

// subscribe订阅render函数，当函数内部状态发生变化时会重新执行render函数

store.subscribe(render) // 状态改变后手动执行render

#### Index.redux.js

const ADD\_GUN = '加机关枪'

const REMOVE\_GUN = '减机关枪'

export function counter(state=0, action) {

switch(action.type) {

case ADD\_GUN:

return state + 1

case REMOVE\_GUN:

return state -1

default:

return 10

}

}

// action creator

export function addGun() {

return { type: ADD\_GUN }

}

export function removeGun() {

return { type: REMOVE\_GUN }

}

// 处理异步

export function addGunAsync() {

return dispatch => {

setTimeout(()=>{

dispatch(addGun())

}, 2000)

}

}

#### App.js

import React from 'react'

class App extends React.Component{

// constructor(props) {

// super(props)

// }

render() {

const store = this.props.store

const num = store.getState()

const addGun = this.props.addGun

const removeGun = this.props.removeGun

const addGunAsync =this.props.addGunAsync

return (

<div className="App">

<h3>现在有机枪{ num }把</h3>

<button onClick={()=>store.dispatch(addGun())}>申请武器</button>

<button onClick={()=>store.dispatch(removeGun())}>上交武器</button>

<button onClick={()=>store.dispatch(addGunAsync())}>拖两天给武器</button>

</div>

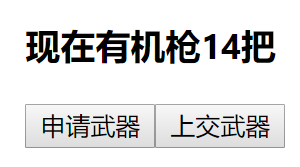
);

}

}

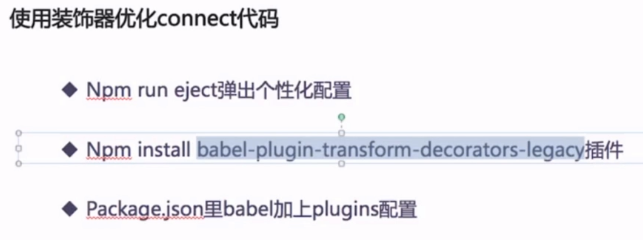
export default App

#### 效果



### Reatc-redux的基本使用(例子)





#### 安装

Yarn add react-redux

#### Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

import { Provider } from 'react-redux'

import { createStore, applyMiddleware, compose } from 'redux'

import thunk from 'redux-thunk'

import { counter } from './index.redux'

// compose把多个中间件连接在一块

const store = createStore(counter, compose(

applyMiddleware(thunk),

window.devToolsExtension? window.devToolsExtension(): f=>f // 使用redux-devtools

))

ReactDOM.render(

<Provider store={ store }>

<App />

</Provider>,

document.getElementById('root')

);

#### Index.redux.js

const ADD\_GUN = '加机关枪'

const REMOVE\_GUN = '减机关枪'

export function counter(state=0, action) {

switch(action.type) {

case ADD\_GUN:

return state + 1

case REMOVE\_GUN:

return state -1

default:

return 10

}

}

// action creator

export function addGun() {

return { type: ADD\_GUN }

}

export function removeGun() {

return { type: REMOVE\_GUN }

}

export function addGunAsync() {

return dispatch => {

setTimeout(()=>{

dispatch(addGun())

}, 2000)

}

}

#### App.js (不使用装饰器)

import React from 'react'

import { connect } from 'react-redux'

import { addGun, removeGun, addGunAsync } from './index.redux'

class App extends React.Component{

render() {

return (

<div className="App">

<h3>现在有机枪{ this.props.num }把</h3>

<button onClick={this.props.addGun}>申请武器</button>

<button onClick={this.props.removeGun}>上交武器</button>

<button onClick={this.props.addGunAsync}>拖两天给武器</button>

</div>

);

}

}

// connect 的四个参数

const mapStatetoProps = (state)=>{

return {num:state}

}

const actionCreators = { addGun, removeGun, addGunAsync }

// 未使用装饰器connect

App = connect(mapStatetoProps, actionCreators)(App)

export default App

#### App.js (使用装饰器)

##### 安装：

yarn add babel-plugin-transform-decorators-legacy

##### package.json

修改babel项

"babel": {

"presets": [

"react-app"

],

"plugins": ["transform-decorators-legacy"]

}

测试后发现以上方法报错，修正为

##### 安装：

yarn add @babel/plugin-proposal-decorators

##### package.json

"babel": {

"presets": [

"react-app"

],

"plugins": [

["@babel/plugin-proposal-decorators", { "legacy": true }]

]

}

##### App.js

import React from 'react'

import { connect } from 'react-redux'

import { addGun, removeGun, addGunAsync } from './index.redux'

// 使用装饰器

@connect(

// 你要state什么属性放到props里

state=>({num:state}),

// 你要什么方法 放到props里 自动dispatch

{ addGun, removeGun, addGunAsync }

)

class App extends React.Component{

render() {

return (

<div className="App">

<h3>现在有机枪{ this.props.num }把</h3>

<button onClick={this.props.addGun}>申请武器</button>

<button onClick={this.props.removeGun}>上交武器</button>

<button onClick={this.props.addGunAsync}>拖两天给武器</button>

</div>

);

}

}

export default App

## Redux-router4

### 安装

yarn add react-router-dom

### 简单使用

#### Index.js

import React from 'react';

import ReactDOM from 'react-dom';

import { Provider } from 'react-redux'

import { createStore, applyMiddleware, compose } from 'redux'

import thunk from 'redux-thunk'

import {

BrowserRouter,

Route,

Link,

Redirect,

Switch

} from 'react-router-dom'

import { counter } from './index.redux'

import App from './App';

// compose把多个中间件连接在一块

const store = createStore(counter, compose(

applyMiddleware(thunk),

window.devToolsExtension? window.devToolsExtension(): f=>f // 使用redux-devtools

))

function Erying() {

return <h2>二营</h2>

}

function Qibinglian() {

return <h2>骑兵连</h2>

}

class Test1 extends React.Component{

render() {

console.log(this.props)

return <h2>{this.props.match.params.location}</h2>

}

}

ReactDOM.render(

<Provider store={ store }>

<BrowserRouter>

<div>

<ul>

<li><Link to='/'>一营</Link></li>

<li><Link to='/erying'>二营</Link></li>

<li><Link to='/qibinglian'>骑兵连</Link></li>

<li><Link to='/dongtai/qibinglian'>动态骑兵连</Link></li>

<li><Link to='/dongtai/qibing'>动态骑兵</Link></li>

</ul>

<Switch> {/\*只渲染命中的第一个Route\*/}

<Route path='/' exact component={App}></Route>

<Route path='/erying' component={Erying}></Route>

<Route path='/qibinglian' component={Qibinglian}></Route>

<Route path='/dongtai/:location' component={Test1}></Route> {/\* 动态路由 \*/}

<Route path='/:location' component={Test1}></Route>

<Redirect to='/qibinglian'></Redirect> {/\* 默认加载的路由 \*/}

</Switch>

</div>

</BrowserRouter>

</Provider>,

document.getElementById('root')

);