# 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>

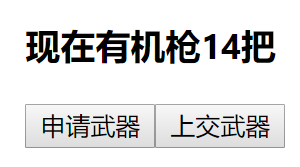
);

}

}

export default App

#### 效果



### Reatc-redux的基本使用(例子)