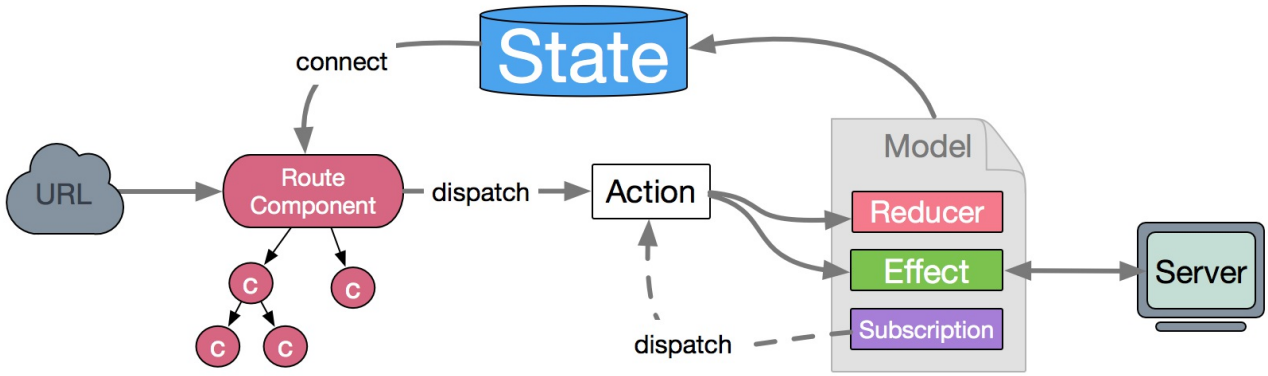
link: null
title: 珠峰架构师成长计划
description: null
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=102 sentences=388, words=3358

1.dva介绍



1.1 前置知识

- react
- react-router-dom
- redux
- react-redux
- connected-react-router
- history
- dva

2. 初始化项目

```
create-react-app zhufeng-dva-source
cd  zhufeng-dva-source
cnpm install dva redux react-redux react-router-dom connected-react-router history
npm start
```

3. 基本计数器

3.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from 'dva';
const app = dva();
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state){
            return {number:state.number+1};
        }
    }
});
const Counter = connect(
    state=>state.counter
)(props=>(
    <div>
        <p>{props.number}p>
        <button onClick={()=>props.dispatch({type:"counter/add"})}>+button>
    div>
));
app.router(()=><Counter/>);
app.start('#root');
```

3.2 src\dva\index.js

src\dva\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import {createStore,combineReducers} from 'redux';
import {connect,Provider} from 'react-redux';
export {connect};
export default function(){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(root){
        let reducers = {};
        for(let i=0;ilet model = app._models[i];
            model.reducers = prefixNamespace(model.reducers,model.namespace);
            reducers[model.namespace] = function(state = model.state, action) {
                let actionType = action.type;
                let reducer = model.reducers[actionType];
                if (reducer) {
                    return reducer(state, action);
                }
                return state;
            };
        }
        let rootReducer = combineReducers(reducers);
        let store = createStore(rootReducer);
        let App = app._router;
        ReactDOM.render(<Provider store={store}><App/>Provider>,document.querySelector(root));
    }
    return app;
}

function prefixNamespace(reducers,namespace){
    return Object.keys(reducers).reduce((memo, key) => {
        const newKey = `${namespace}${NAMESPACE_SEPARATOR}${key}`;
        memo[newKey] = reducers[key];
        return memo;
    }, {});
}
```

4. 支持effects

4.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from 'dva';
const app = dva();
+ function delay(ms) {
+    return new Promise((resolve,reject) => {
+        setTimeout(function () {
+            resolve();
+        },ms);
+    });
+ }
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state){
            return {number:state.number+1};
        }
    },
+    effects:{
+        *asyncAdd(action,{call,put}){
+          yield call(delay,1000);
+          yield put({type:'counter/add'});
+        }
+    }
});
const Counter = connect(
    state=>state.counter
)(props=>(

        {props.number}
        props.dispatch({type:"counter/add"})}>+
+        props.dispatch({type:"counter/asyncAdd"})}>异步+

));
app.router(()=>);
app.start('#root');
```

4.2 src\dva\index.js

src\dva\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
+import {createStore,combineReducers,applyMiddleware} from 'redux';
import {connect,Provider} from 'react-redux';
+import createSagaMiddleware from 'redux-saga';
+import * as sagaEffects from 'redux-saga/effects';
export {connect};
export default function(){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(root){
        let reducers = {};
        for(let i=0;i+        let sagaMiddleware = createSagaMiddleware();
+        function* rootSaga(){
+            let {takeEvery} =  sagaEffects;
+            for(const model of app._models){
+                for(const key in model.effects){
+                    yield takeEvery(model.namespace+'/'+key,function*(action){
+                        yield model.effects[key](action,sagaEffects);
+                    });
+                }
+            }
+        }
+        let store = createStore(rootReducer,applyMiddleware(sagaMiddleware));
+        sagaMiddleware.run(rootSaga);
        let App = app._router;
        ReactDOM.render(,document.querySelector(root));
    }
    return app;
}
```

5.支持路由

5.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from 'dva';
+import {Router,Route} from 'dva/router';
const app = dva();
function delay(ms) {
    return new Promise((resolve,reject) => {
        setTimeout(function () {
            resolve();
        },ms);
    });
}
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state){
            return {number:state.number+1};
        }
    },
    effects:{
        *asyncAdd(action,{call,put}){
          yield call(delay,1000);
          yield put({type:'counter/add'});
        }
    }
});
const Counter = connect(
    state=>state.counter
)(props=>(

        {props.number}
        props.dispatch({type:"counter/add"})}>+
        props.dispatch({type:"counter/asyncAdd"})}>异步+

));
//index.js:1375 Warning: Please use `require("history").createHashHistory` instead of `require("history/createHashHistory")`. Support for the latter will be
removed in the next major release.

//index.js:1375 Warning: [sagaEffects.put] counter/add should not be prefixed with namespace counter
+const Home = ()=>Home
+app.router(({history})=>(
+
+        <>
+
+
+        </>
+
+));
app.start('#root');
```

5.2 src\dva\index.js

src\dva\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import {createStore,combineReducers,applyMiddleware} from 'redux';
import {connect,Provider} from 'react-redux';
import createSagaMiddleware from 'redux-saga';
import * as sagaEffects from 'redux-saga/effects';
+import {createHashHistory} from 'history';
export {connect};
export default function(){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(root){
        let reducers = {};
        for(let i=0;i+        let history = createHashHistory();
+        let App = app._router({history});
+        ReactDOM.render({App},document.querySelector(root));
    }
    return app;
}
```

5.3 src\dva\router.js

src\dva\router.js

```
export * from 'react-router-dom';
```

6.支持跳转

6.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from './dva';
+ import {Router,Route,routerRedux} from './dva/router';
const app = dva();
function delay(ms) {
    return new Promise((resolve,reject) => {
        setTimeout(function () {
            resolve();
        },ms);
    });
}
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state){
            return {number:state.number+1};
        }
    },
    effects:{
        *asyncAdd(action,{call,put}){
          yield call(delay,1000);
          yield put({type:'counter/add'});
        },
+       *goto({to}, { put }) {
+         yield put(routerRedux.push(to));
+       }
    }
});
const Counter = connect(
    state=>state.counter
)(props=>(

        {props.number}
        props.dispatch({type:"counter/add"})}>+
        props.dispatch({type:"counter/asyncAdd"})}>异步+
+       props.dispatch({type:"counter/goto",to:'/'})}>跳转到/

));
//index.js:1375 Warning: Please use `require("history").createHashHistory` instead of `require("history/createHashHistory")`. Support for the latter will be
removed in the next major release.

//index.js:1375 Warning: [sagaEffects.put] counter/add should not be prefixed with namespace counter
const Home = ()=>Home
app.router(({history})=>(
        <>

        </>
));
app.start('#root');
```

6.2 src\dva\index.js

src\dva\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import {createStore,combineReducers,applyMiddleware} from 'redux';
import {connect,Provider} from 'react-redux';
import createSagaMiddleware from 'redux-saga';
import * as sagaEffects from 'redux-saga/effects';
import {createHashHistory} from 'history';
+import {
+  routerMiddleware,
+  connectRouter,
+  ConnectedRouter
+} from "connected-react-router";
+const NAMESPACE_SEPARATOR = "/";
export {connect};
export default function(){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(root){
+       let history = createHashHistory();
+       const reducers = {
+           router: connectRouter(history)
+       };
        for(let i=0;i+                    yield takeEvery(model.namespace+NAMESPACE_SEPARATOR+key,function*(action){
+                        yield model.effects[key](action,sagaEffects);
+                    });
                }
            }
        }
+       let store = createStore(rootReducer,applyMiddleware(routerMiddleware(history),sagaMiddleware));
        sagaMiddleware.run(rootSaga);
+       let App = app._router({history,app});
+       ReactDOM.render(
+
+
+             {App}
+
+       ,document.querySelector(root));
    }
    return app;
}
```

6.3 src\dva\router.js

src\dva\router.js

```
+module.exports = require('react-router-dom');
+module.exports.routerRedux = require('connected-react-router');
```

7.支持自定义effects

7.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from 'dva';
import {Router,Route,routerRedux} from 'dva/router';
const app = dva();
function delay(ms) {
    return new Promise((resolve,reject) => {
        setTimeout(function () {
            resolve();
        },ms);
    });
}
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state,){
            return {number:state.number+1};
        },
        increment(state,{payload}){
           return {number:state.number+payload||1};
        }
    },
    effects:{
       *asyncAdd(action,{call,put}){
          yield call(delay,1000);
          yield put({type:'counter/add'});
       },
       *goto({to}, { put }) {
         yield put(routerRedux.push(to));
       },
+       addWatchers:[
+           function*({take,put,call}){
+               for(let i=0;i
+                   const {payload} = yield take('counter/addWatcher');
+                   yield call(delay,1000);
+                   yield put({type:'counter/increment',payload});
+               }
+               alert('不能再加了');
+           },
+           {type:'watcher'}
+       ]
    }
});
const Counter = connect(
    state=>state.counter
)(props=>(

        {props.number}
        props.dispatch({type:"counter/add"})}>+
        props.dispatch({type:"counter/asyncAdd"})}>异步+
+        props.dispatch({type:"counter/addWatcher",payload:2})}>自定义异步+2
        props.dispatch({type:"counter/goto",to:'/'})}>跳转到/

));
const Home = ()=>Home
app.router(({history})=>(

        <>

        </>

));
app.start('#root');
```

7.2 src\dva\index.js #

```
import React from 'react';
import ReactDOM from 'react-dom';
import {createStore,combineReducers,applyMiddleware} from 'redux';
import {connect,Provider} from 'react-redux';
import createSagaMiddleware from 'redux-saga';
import * as sagaEffects from 'redux-saga/effects';
import {createHashHistory} from 'history';
import {
  routerMiddleware,
  connectRouter,
  ConnectedRouter
} from "connected-react-router";
export {connect};
export default function(){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(root){
        let history = createHashHistory();
        const reducers = {
            router: connectRouter(history)
        };
        for(let i=0;i+                for(const key in model.effects){
+                   let effect = model.effects[key];
+                   if(Array.isArray(effect) && effect[1].type == 'watcher'){
+                      yield sagaEffects.fork(effect[0],sagaEffects);
+                   }else{
+                     yield takeEvery(model.namespace+'/'+key,function*(action){
+                       yield model.effects[key](action,sagaEffects);
+                     });
+                   }
+               }
+           }
        }
        let store = createStore(rootReducer,applyMiddleware(routerMiddleware(history),sagaMiddleware));
        sagaMiddleware.run(rootSaga);
        let App = app._router({history,app});
        ReactDOM.render(

             {App}

        ,document.querySelector(root));
    }
    return app;
}
```

8.支持钩子

8.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from 'dva';
import {Router,Route,routerRedux} from 'dva/router';
+import { message } from 'antd';
+import logger from 'redux-logger';
+import {createBrowserHistory} from 'history';
+import 'antd/dist/antd.css'
+const SHOW = 'SHOW';
+const HIDE = 'HIDE';
+const initialState = {
+    global: false,
+    models: {},
+    effects: {}
+};
+const app = dva({
+    history:createBrowserHistory(),//自定义history
+    initialState:{counter:{number:5}},//自定义初始状态
+    //effect 执行错误或 subscription 通过 done 主动抛错时触发，可用于管理全局出错状态
+    onError:((err, dispatch) => {message.error(err.message)}),
+    //在 action 被 dispatch 时触发，用于注册 redux 中间件。支持函数或函数数组格式
+    onAction:logger,
+    //state 改变时触发，可用于同步 state 到 localStorage，服务器端等
+    onStateChange:(state)=>{console.log(state)},
+    //封装 reducer 执行。比如借助 redux-undo 实现 redo/undo
+    onReducer:reducer=>(state,action)=>{//增加额外的reducer
+      localStorage.setItem('action',JSON.stringify(action));
+      return reducer(state,action);
+    },
+    //封装 effect 执行。比如 dva-loading 基于此实现了自动处理 loading 状态。
+    onEffect:(effect, { put }, model, actionType)=>{
+      const { namespace } = model;
+      return function*(...args) {
+        yield put({ type: SHOW, payload: { namespace, actionType } });
+        yield effect(...args);
+        yield put({ type: HIDE, payload: { namespace, actionType } });
+      };
+    },
+    //指定额外的 reducer，比如 redux-form 需要指定额外的 form reducer:
+    extraReducers:{
+      loading(state = initialState, { type, payload }) {
+        const { namespace, actionType } = payload || {};
```

```
+        switch(type){
+          case SHOW:
+            return {global:true,models: { ...state.models, [namespace]: true }, effects: { ...state.effects, [actionType]: true },};
+          case HIDE:
+            const effects = { ...state.effects, [actionType]: false };
+            const models = { ...state.models, [namespace]: false };
+            const global = Object.keys(models).some(namespace => {
+                return models[namespace];
+            });
+            return {global,models,effects};
+          default:
+            return state;
+        }
+      }
+    },
+    //指定额外的 StoreEnhancer ，比如结合 redux-persist 的使用:
+    //指定额外的 StoreEnhancer
+    //它的参数是创建store的函数（store creator），返回值是一个可以创建功能更加强大的store的函数(enhanced store creator)
+    extraEnhancers:(createStore)=>{
+       return (...args)=>{
+           let store = createStore(...args);
+           console.log('返回更强大的store')
+           return {...store,more(){console.log('更强大的强能')}};
+       }
+    }
+});
function delay(ms) {
    return new Promise((resolve,reject) => {
        setTimeout(function () {
            resolve();
        },ms);
    });
}
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state,){
            return {number:state.number+1};
        },
        increment(state,{payload}){
           return {number:state.number+payload||1};
        }
    },
    effects:{
       *asyncAdd(action,{call,put}){
          yield call(delay,1000);
          yield put({type:'counter/add'});
       },
       *goto({to}, { put }) {
         yield put(routerRedux.push(to));
       },
       addWatchers:[
           function*({take,put,call}){
               for(let i=0;istate.counter
)(props=>(

        {props.number}
        props.dispatch({type:"counter/add"})}>+
        props.dispatch({type:"counter/asyncAdd"})}>异步+
        props.dispatch({type:"counter/addWatcher",payload:2})}>自定义异步+2
        props.dispatch({type:"counter/goto",to:'/'})}>跳转到/

));
const Home = ()=>Home
app.router(({history})=>(

        <>

        </>

));
app.start('#root');
```

8.2 src\dva\index.js

src\dva\index.js

```
import React from "react";
import ReactDOM from "react-dom";
import { createStore, combineReducers, applyMiddleware } from "redux";
import { connect, Provider } from "react-redux";
import createSagaMiddleware from "redux-saga";
import * as sagaEffects from "redux-saga/effects";
import { createHashHistory } from "history";
import {
  routerMiddleware,
  connectRouter,
  ConnectedRouter
} from "connected-react-router";
const NAMESPACE_SEPARATOR = "/";
export { connect };
export default function(options) {
  const app = {
    _models: [],
    model,
    _router: null,
    router,
    start
  };
  function model(model) {
    app._models.push(model);
  }
  function router(routerConfig) {
```

```
    app._router = routerConfig;
  }
+  let history = options.history||createHashHistory();
  function start(root) {
-   let history = createHashHistory();
+    let reducers = {
+      router: connectRouter(history)
+    };
+    if(options.extraReducers){
+        reducers = {...reducers,...options.extraReducers};
+    }
    for (let i = 0; i < app._models.length; i++) {
      let model = app._models[i];
      reducers[model.namespace] = function(state = model.state, action) {
        let actionType = action.type;
        let values = actionType.split(NAMESPACE_SEPARATOR);
        if (values[0]
          let reducer = model.reducers[values[1]];
          if (reducer) {
            return reducer(state, action);
          }
        }
        return state;
      };
    }
+    let combinedReducer  = combineReducers(reducers);
+    let rootReducer = function(state,action){
+        let newState =  combinedReducer(state,action);
+        options.onStateChange&&options.onStateChange(newState);
+        return newState;
+    }
+    if(options.onReducer){
+        rootReducer = options.onReducer(rootReducer);
+    }
    let sagaMiddleware = createSagaMiddleware();
    function* rootSaga() {
      let { takeEvery } = sagaEffects;
      for (const model of app._models) {
        for (const key in model.effects) {
          let effect = model.effects[key];
          if (Array.isArray(effect) && effect[1].type == "watcher") {
            yield sagaEffects.fork(effect[0], sagaEffects);
          } else {
            yield takeEvery(
              model.namespace + NAMESPACE_SEPARATOR + key,
              function*(action) {
                yield model.effects[key](action, sagaEffects);
              }
            );
          }
        }
      }
    }
+    if(options.onAction){
+        if(typeof options.onAction == 'function'){
+            options.onAction = [options.onAction];
+        }
+    }else{
+        options.onAction=[];
+    }
+    let enhancedCreateStore;
+    if(options.extraEnhancers){
+        enhancedCreateStore = options.extraEnhancers(createStore);
+    }
+    let store = enhancedCreateStore(
+      rootReducer,
+      applyMiddleware(routerMiddleware(history), sagaMiddleware,...options.onAction)
+    );
    sagaMiddleware.run(rootSaga);
    let App = app._router({ history, app });
    ReactDOM.render(

        {App}
      ,
      document.querySelector(root)
    );
  }
  return app;
}
```

8.版本

8.1 src\index.js

src\index.js

```
import React from 'react';
import dva,{connect} from './dva';
import {Router,Route,routerRedux} from './dva/router';
import {createBrowserHistory} from 'history';
import {message} from 'antd';
import 'antd/dist/antd.css';
import logger from 'redux-logger';
const SHOW = 'SHOW';
const HIDE = 'HIDE';
const initialLoadingState = {
    global:false,//只有有任何一个saga执行的它就是为true,只有所有的saga都不执行它才为false
    models:{},//里面放着哪个model有saga正在执行
    effects:{},//model中的哪个saga正在执行
}
//routerRedux= connected-react-redux导出对象
const app = dva({
    history:createBrowserHistory(),//指定用哪个历史对象
```

```
    initialState:{counter:{number:5}}, //初始状态
    //当effect执行错误会走到这里
    onError:(err,dispatch)=>{message.error(err.toString())},
    //值可能是一个中间件函数，也可能是一个中间件的数组
    onAction:logger,
    //当仓库中状态发生改变的时候，执行此函数
    onStateChange:state=>console.log('onStateChange',state),
    //onReducer 是用来增强reducer的
    onReducer:reducer=>(state,action)=>{
        let newState =  reducer(state,action);
        //newState.counter.number*=2;
        return newState;
    },
    //封闭effect的执行 dva-loading 老的saga effects model 动作类型
    onEffect:(effect,{put},model,actionType)=>{
        const {namespace} = model;
        return function*(...args){
            yield put({type:SHOW,payload:{namespace,actionType}});//namespace=counter actionType asyncAdd
            yield effect(...args);
            yield put({type:HIDE,payload:{namespace,actionType}});
        }
    },
    //额外的reducer对象
    extraReducers:{
        //当向仓库派发动作的时候
        loading(state=initialLoadingState,{type,payload}){
            const {namespace,actionType} = payload||{};//namespace =counter actionType -counter/asyncAdd
            switch(type){
                case SHOW:
                  return {
                      global:true,//只要有一个effect执行了，它就为true
                      models:{...state.models,[namespace]:true},
                      effects:{...state.effects,[actionType]:true}
                  }
                case HIDE:
                  const effects =  {...state.effects,[actionType]:false};
                  const models = {...state.models,[namespace]:false};
                  const global = Object.keys(models).some(namespace=>models[namespace]);
                  return {effects,models,global};
                default:
                  return state;
            }

        }
    },
    //额外的增强器  用来增强createStore
    extraEnhancers:createStore=>{
        return (...args)=>{
            let store = createStore(...args);
            console.log('我正在创建仓库');
            return store;
        }
    }
});
const delay = ms=>new Promise(function(resolve){
    setTimeout(() => {resolve();}, ms);
});
app.model({
    namespace:'counter',
    state:{number:0},
    reducers:{
        add(state,{payload}){
            return {number:state.number+(payload||1)};
        }
    },
    effects:{
        *asyncAdd(action,{put,call}){
            yield call(delay,1000);
            throw new Error('异步加1失败');
            yield put({type:'counter/add'});
        },
        *goto({payload},{put}){
            yield put(routerRedux.push(payload));//connected-react-router
        },
        addWatcher:[
            function*({take,put,call}){
                for(let i=0;istate.counter
)(
  props=>(

        {props.number}
        props.dispatch({type:'counter/add'})}>+
        props.dispatch({type:'counter/asyncAdd'})}>异步加1
        props.dispatch({type:'counter/addWatcher'})}>addWatcher
        props.dispatch({type:'counter/goto',payload:'/'})}>跳转到/路径里

  )
);
const Home = props=>首页;
app.router(({history})=>(

      <>

      </>

));
app.start('#root');
```

8.2 dva\index.js

src\dva\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import {createStore,combineReducers,applyMiddleware} from 'redux';
import {connect,Provider} from 'react-redux';
import createSagaMiddleware from 'redux-saga';
import * as effects from 'redux-saga/effects';
import createRootSaga from './createRootSaga';
import createReducers from './createReducers';
import {createHashHistory} from 'history';
import {routerMiddleware,connectRouter,ConnectedRouter} from 'connected-react-router';
import {prefix} from './utils';
export {connect}

export default function(opts){
    const app = {
        _models:[],
        model,
        _router:null,
        router,
        start
    }
    function model(model){
        app._models.push(model);
    }
    function router(routerConfig){
        app._router = routerConfig;
    }
    function start(selector){
        let history = opts.history||createHashHistory();
        let rootReducer = createReducers(app._models,history,opts.extraReducers);
        let finalReducer = function(state,action){
            let newRootReducer = opts.onReducer(rootReducer);
            let newState = newRootReducer(state,action);
            if(opts.onStateChange){
                 opts.onStateChange(newState);
            }
            return newState;
        }
        let sagaMiddleware = createSagaMiddleware();
        let rootSaga = createRootSaga(app._models,opts.onError||function(){},opts.onEffect);
        if(opts.onAction){
            if(!Array.isArray(opts.onAction)){
                opts.onAction=[opts.onAction];
            }
        }else {
         opts.onAction=[]
        }
        let newCreateStore = createStore;
        if(opts.extraEnhancers){
            newCreateStore = opts.extraEnhancers(createStore);
        }
        let store = newCreateStore(finalReducer,opts.initialState||undefined,
        applyMiddleware(routerMiddleware(history),sagaMiddleware,...opts.onAction));
        sagaMiddleware.run(rootSaga);
        let App = app._router({history});
        ReactDOM.render(
            <Provider store={store}>
                <ConnectedRouter history={history}>
                    {App}
                ConnectedRouter>
            Provider>
            ,document.querySelector(selector));
    }
    return app;
}
```

8.3 router.js

src\dva\router.js

```
module.exports = require('react-router-dom');
module.exports.routerRedux = require('connected-react-router');
```

8.4 createReducers.js

src\dva\createReducers.js

```
import {combineReducers} from 'redux';
import {prefix} from './utils';
import {connectRouter} from 'connected-react-router';
export default function createReducers(models,history,extraReducers) {
  let reducers = {router:connectRouter(history),...extraReducers};
  for (let model of models) {
    let { namespace, state: initialState, reducers: modelReducers } = model;
    let reducersWithPrefix = prefix(modelReducers, namespace);
    reducers[namespace] = function(state = initialState, action) {
      let reducer = reducersWithPrefix[action.type];
      if (reducer) {
        return reducer(state, action);
      }
      return state;
    };
  }
  return  combineReducers(reducers);
}
```

8.5 createRootSaga.js

src\dva\createRootSaga.js

```
import {prefix} from './utils';
import * as sagaEffects from 'redux-saga/effects';
export default function createRootSaga(models,onError,onEffect){
   function* rootSaga(){
       try{
         for(let model of models){
           model.effects = prefix(model.effects,model.namespace);
           for(let key in model.effects){
              let effect = model.effects[key];
              if(Array.isArray(effect)&&effect[1].type == 'watcher'){
                  let effect = model.effects[key];
                  yield sagaEffects.fork(effect[0],sagaEffects);
              }else{
                yield sagaEffects.takeEvery(key,function*(action){
                    let oldEffect = model.effects[key];
                    let newEffect = onEffect(oldEffect,sagaEffects,model,action.type);
                    yield newEffect(action,sagaEffects);
                });
              }
           }
         }
       }catch(error){
           onError(error);
       }

   }
   return rootSaga;
}
```

8.参考

- dvajs (https://dvajs.com/)
- zhufeng-dva-source4 (https://gitee.com/zhufengpeixun/zhufeng-dva-source4.git)