讲师管理系统

1 开发流程

1.1 第一天

1. 初始化项目

```
create-react-app teacher_manager
```

2. 使用dva 引入History 和 dva

```
npm i history dva -S
```

3. 改造`src/index.js` 通过dva的方式启动项目

```
// import React from 'react';
// import ReactDOM from 'react-dom';
// 使用history_1: 引入createHistory
import { createBrowserHistory  } from 'history'

// 引入样式 使用 bootstrap
import 'bootstrap/dist/css/bootstrap.min.css'
import 'font-awesome/css/font-awesome.css'
import './style/index.css'

import './index.css';
// import App from './App';
import * as serviceWorker from './serviceWorker';

// 通过dva 的方式启动项目
// 1、引入DVA
import Dva from 'dva'
// 默认查找index首页
import router from './routers'
import model from './models'

// 2、创建dva的实例，APP对象
// 使用history_2: 传入Dva
const app = new Dva({
  history: createBrowserHistory()
})

// 3、配置路由app.router(fn)
app.router(router)

// 4、app.use();  // 安装插件

// 5、注册模块 app.model
app.model(model)

```

```
// 5.5、 卸载模块 app.unmodel()

// 6、app.start('#root)  // root  就是根节点的id
app.start('#root')


serviceWorker.unregister();


```

4. 编写`src/router/index.js`

```

// 配置路由规则
import React from 'react'
import { Router, Route, NavLink, Switch, Redirect } from 'dva/router'

const Home = () => <h1>您好</h1>

// 使用history 需要先进行声明 在 src/index.js 中传入 history 【转到 src/index.js查看具体步骤】
// history 没有 # 号
let fn = function ({ history, app }) {
  return (
    <Router history={ history }>
      <React.Fragment>
          // exact  不做嵌套路由
        <Route path="/home" exact component={ Home } />
      </React.Fragment>
    </Router>
  )
}


// 导出为函数
export default fn;
```

5. 编写`src/model/index.js`

```
let index = {
  // 命名空间 具体命名
  namespace: 'index',
  state: {
    num: 0
  },
  effects: {
    // 异步action
    /* payload: 发来的action 中携带的数据
    select： 获取当前的state
    put: 调用reducers
    call： 调用异步方法 */
    *changeNum ({ payload }, { select, put, call }) {
      // 保持reducers 中名字的一致性
      console.log(payload)
      yield put({ type: 'addNum', payload })
    }
  },
  reducers: {
    /*
      let two = {
        payload: {num: 8}
        type: "index/addNum"
      }
      let { payload } = two  // { num: 8 }
    */
    addNum (state, { payload }) {
      // 返回一个新对象 值不可变性
      console.log(state, payload)
      return {
        num: state.num + payload.num
      }
    }
  }
}


export default index
```

6. 使用store （connect）新建`src/components/Home.js`

```
import React, { Component } from 'react'
// 使用connect_1 引入
import { connect } from 'dva'

class Home extends Component {
  render () {
    console.log(this.props.num)
    return (
      <React.Fragment>
        <h1>
          数量: { this.props.num }
        </h1>
        <button onClick={ e=> {
          // 带上命名空间 ，并传递payload
          this.props.dispatch({ type: 'index/changeNum', payload: { num:8 } })
        } }>
          增加number
        </button>
      </React.Fragment>
    )
  }
}

export default connect(state => {
  console.log(state)
  // 使用connect_2 声明在prop中需要使用的
  return {
    // 通过命名去state 中去取
    num: state.index.num
  }
})(Home)

```

7. 使用 axios

```
npm i axios -S
```

   1. 封装请求

      `src/config.js`

      ```
      export const serverHost = 'http://www.sinya.online'
      export const port = 3002
      ```

   2. `src/utils/request.js`

      ```
      import Axios from 'axios'
      import { serverHost, port } from '../config/index'

      let r = Axios.create({
        baseURL: `${serverHost}:${port}/`
      })

      let request = function (url='', options={}) {
        if (url === '') {
          return Promise.reject('必须传递url')
        }
        return r({
          url,
          method: 'get',
          ...options
        })
      }

      export default request
      ```

1.2 登录

1. `src/components/Signin.js`

```
import React, { Component } from 'react'

// 处理图片 手动告知webPack 加载图片
import monkey from '../assets/img/monkey.png'

class Signin extends Component {
  constructor () {
    super()
    this.state = {
      username: '',
      password: ''
    }
  }

  // 调用teacher中的登录异步方法
  handelSignin (e) {
    e.preventDefault()
    let { username, password } = this.state
```

```
    console.log(username)
    this.props.dispatch({
      type: 'teacher/doLogin',
      payload: {
        username,
        password
      }
    })
  }
  render () {
    return (
      <React.Fragment>
              <form action=""  className="col-xs-offset-1 col-xs-10" onSubmit={ e=> this.handelSignin() }>
                  <input id="name" type="text" className="form-control" placeholder="用户名"
                    value={ this.state.username } onChange={e=> {
                      this.setState({
                        username: e.target.value
                      })
                    }}
                  />
                  <input id="pass" type="password" className="form-control" placeholder="密码"
                    value={ this.state.password } onChange={e=> {
                      this.setState({
                        password: e.target.value
                      })
                    }}
                  />
                />
      </React.Fragment>
    )
  }
}


export default connect(state=> {
  return {
    isLogin: state.teacher.isLogin
  }
})(Signin)
```

2. `src/routers/index.js`

   在路由中注册

```
import Signin from '../components/Signin'

// 使用history 需要先进行声明
let fn = function ({ history, app }) {
  return (
    <Router history={ history }>
      <React.Fragment>
        <Route path="/home" exact component={ Home } />
        <Route path="/signin" exact component={ Signin } />
      </React.Fragment>
    </Router>
  )
}


// 导出为函数
export default fn;
```

3. 编写并注册`teacher`模块 在 `src/models/teacher.js`

   1. 编写 使用 call 函数

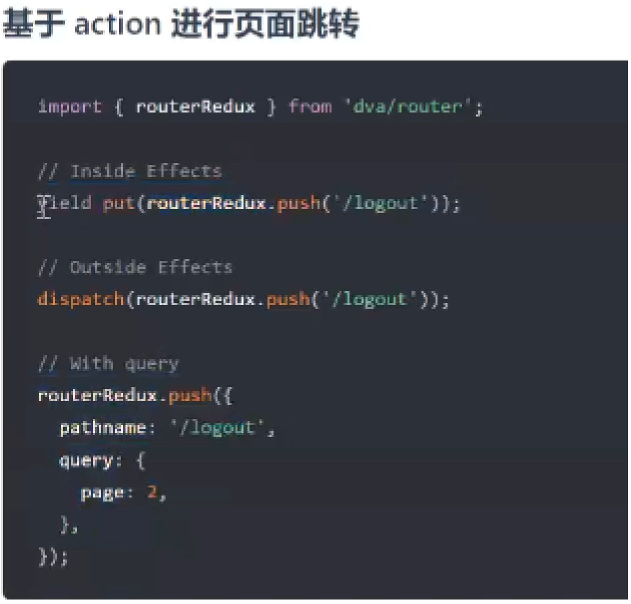
```
import request from '../utils/request'
// 做路由的跳转使用
import { routerRedux } from 'dva/router'

import * as api from '../api/index'

let teacher = {
  // 命名空间 具体命名
  namespace: 'teacher',
  state: {
    isLogin: false
  },
  effects: {
    // 异步action 同步返回 结果
    /* payload: 发来的action 中携带的数据
    select:  获取当前的state
    put: 调用reducers
    call:  调用异步方法 */
    *doLogin ({ payload }, { select, put, call }) {
      let res
      try {
        // call 参数为一个函数,这个函数里面返回一个函数，此函数返回一个Promise
        // 同步返回
        /*
            let request = function (url='', options={}) {
              return function () {
                if (url === '') {
                  return Promise.reject('必须传递url')
                }
                return r({
                  url,
                  method: 'get',
                  ...options
                })
              }
            }
            res = yield call(request(‘signin’， payload))
        */
        res = yield call(() => api.login(payload))
   } catch (error) {
        console.log('登录失败', error)
   }
      if (res.data.code === 0) {
        console.log(res.data)
        window.sessionStorage.setItem('user', JSON.stringify(res.data.data.user))
        yield put({type: 'changeLogin', payload: {isLogin: true}})
         //
        yield put(routerRedux.push('/home'))
      }
    }
  },
  reducers: {
 /*
      整个对象是 action  只要action中的payload
     action:  {
        payload: {num: 8}
        type: "index/addNum"
      }
      let { payload } = two  // { num: 8 }
    */
    changeLogin (state, { payload }) {
      return {
        isLogin: payload.isLogin
      }
    }
  }
}


export default teacher
```

在effects中进行路由的跳转



基于 action 进行页面跳转

```
import { routerRedux } from 'dva/router';

// Inside Effects
yield put(routerRedux.push('/logout'));

// Outside Effects
dispatch(routerRedux.push('/logout'));

// With query
routerRedux.push({
  pathname: '/logout',
  query: {
    page: 2,
  },
});
```

2. 在 `src/index.js` 中注册

```
// 注册teacher model
import teacher from './models/teacher'

// 注册 teacher 模块
app.model(teacher)
```

1.3 首页

1.3.1 重定向到登录页

`src/routers/index`

```
    <Router history={ history }>
      <React.Fragment>
        <Switch>
          <Route path="/home" component={ Home } />
          <Route path="/signin" exact component={ Signin } />
           // 重定向 外层需要有Switch 做横向的匹配
          <Redirect to="/signin" />
        </Switch>
      </React.Fragment>
    </Router>
```

1.3.2 首页嵌套路由

`src/routers/index`

```
export const MyRoute = Route;
// 首页的嵌套路由表
export const homeSubRouters = [
  {
    path: '/home/list',
    component: TeacherList
  }
]

...
// 嵌套路由不能有exact(精确匹配) 关键字
// 因为如果做精确匹配的话 父组件就显示不出来
<Route path="/home" component={ Home } />
...
```

`src/components/Home.js`

```
import { homeSubRouters, MyRoute as Route, MyRouter as Router } from '../routers';

{/* 嵌套路由 */}
{
    homeSubRouters.map((route, i)=>{
        return <Route key={i} path={route.path} component={route.component}/>
    })
}
```

1.4 使用Loading

1.4.1 全局Loading

1. 安装`dva-loading`

```
npm i dva-loading -S
```

2. 安装插件`src/index.js`

```
import createLoading from 'dva-loading';

// 4、app.use();  // 安装插件
app.use(createLoading({
  // 创建了一个命名空间
  namespace:'myloading', //默认是loading没有啥用
}));
```

3. 创建`Loading.css`和`Loading.js`
   1. `Loading.css`

```
.windows8 {
    position: relative;
    width: 78px;
    height:78px;
    margin:auto;
}

.windows8 .wBall {
    position: absolute;
    width: 74px;
    height: 74px;
    opacity: 0;
    transform: rotate(225deg);
        -o-transform: rotate(225deg);
        -ms-transform: rotate(225deg);
        -webkit-transform: rotate(225deg);
        -moz-transform: rotate(225deg);
    animation: orbit 6.96s infinite;
        -o-animation: orbit 6.96s infinite;
        -ms-animation: orbit 6.96s infinite;
        -webkit-animation: orbit 6.96s infinite;
        -moz-animation: orbit 6.96s infinite;
}

.windows8 .wBall .wInnerBall{
    position: absolute;
    width: 10px;
    height: 10px;
    background: rgb(0,0,0);
    left:0px;
    top:0px;
    border-radius: 10px;
}

.windows8 #wBall_1 {
    animation-delay: 1.52s;
        -o-animation-delay: 1.52s;
        -ms-animation-delay: 1.52s;
        -webkit-animation-delay: 1.52s;
        -moz-animation-delay: 1.52s;
}

.windows8 #wBall_2 {
    animation-delay: 0.3s;
        -o-animation-delay: 0.3s;
        -ms-animation-delay: 0.3s;
        -webkit-animation-delay: 0.3s;
        -moz-animation-delay: 0.3s;
}

.windows8 #wBall_3 {
    animation-delay: 0.61s;
        -o-animation-delay: 0.61s;
        -ms-animation-delay: 0.61s;
        -webkit-animation-delay: 0.61s;
        -moz-animation-delay: 0.61s;
}

.windows8 #wBall_4 {
    animation-delay: 0.91s;
        -o-animation-delay: 0.91s;
        -ms-animation-delay: 0.91s;
        -webkit-animation-delay: 0.91s;
        -moz-animation-delay: 0.91s;
}

.windows8 #wBall_5 {
    animation-delay: 1.22s;
        -o-animation-delay: 1.22s;
        -ms-animation-delay: 1.22s;
        -webkit-animation-delay: 1.22s;
        -moz-animation-delay: 1.22s;
}



@keyframes orbit {
    0% {
        opacity: 1;
        z-index:99;
        transform: rotate(180deg);
        animation-timing-function: ease-out;
    }

    7% {
        opacity: 1;
        transform: rotate(300deg);
        animation-timing-function: linear;
        origin:0%;
    }

    30% {
        opacity: 1;
        transform:rotate(410deg);
        animation-timing-function: ease-in-out;
        origin:7%;
    }

    39% {
        opacity: 1;
        transform: rotate(645deg);
        animation-timing-function: linear;
        origin:30%;
    }

    70% {
        opacity: 1;
        transform: rotate(770deg);
        animation-timing-function: ease-out;
        origin:39%;
    }

    75% {
        opacity: 1;
        transform: rotate(900deg);
        animation-timing-function: ease-out;
        origin:70%;
    }

    76% {
    opacity: 0;
        transform:rotate(900deg);
    }

    100% {
    opacity: 0;
        transform: rotate(900deg);
    }
}

@-o-keyframes orbit {
    0% {
        opacity: 1;
        z-index:99;
        -o-transform: rotate(180deg);
        -o-animation-timing-function: ease-out;
    }

    7% {
        opacity: 1;
        -o-transform: rotate(300deg);
        -o-animation-timing-function: linear;
        -o-origin:0%;
    }

    30% {
        opacity: 1;
        -o-transform:rotate(410deg);
        -o-animation-timing-function: ease-in-out;
        -o-origin:7%;
    }

    39% {
        opacity: 1;
        -o-transform: rotate(645deg);
        -o-animation-timing-function: linear;
        -o-origin:30%;
    }

    70% {
        opacity: 1;
        -o-transform: rotate(770deg);
        -o-animation-timing-function: ease-out;
        -o-origin:39%;
    }

    75% {
        opacity: 1;
        -o-transform: rotate(900deg);
        -o-animation-timing-function: ease-out;
        -o-origin:70%;
    }

    76% {
    opacity: 0;
        -o-transform:rotate(900deg);
    }

    100% {
    opacity: 0;
        -o-transform: rotate(900deg);
    }
}

@-ms-keyframes orbit {
    0% {
        opacity: 1;
        z-index:99;
        -ms-transform: rotate(180deg);
        -ms-animation-timing-function: ease-out;
    }

    7% {
        opacity: 1;
        -ms-transform: rotate(300deg);
        -ms-animation-timing-function: linear;
        -ms-origin:0%;
    }

    30% {
        opacity: 1;
        -ms-transform:rotate(410deg);
        -ms-animation-timing-function: ease-in-out;
        -ms-origin:7%;
    }

    39% {
        opacity: 1;
        -ms-transform: rotate(645deg);
        -ms-animation-timing-function: linear;
        -ms-origin:30%;
    }

    70% {
        opacity: 1;
        -ms-transform: rotate(770deg);
        -ms-animation-timing-function: ease-out;
        -ms-origin:39%;
    }

    75% {
        opacity: 1;
        -ms-transform: rotate(900deg);
        -ms-animation-timing-function: ease-out;
        -ms-origin:70%;
    }

    76% {
    opacity: 0;
        -ms-transform:rotate(900deg);
    }

    100% {
    opacity: 0;
        -ms-transform: rotate(900deg);
    }
}

@-webkit-keyframes orbit {
    0% {
        opacity: 1;
        z-index:99;
        -webkit-transform: rotate(180deg);
        -webkit-animation-timing-function: ease-out;
    }

    7% {
        opacity: 1;
        -webkit-transform: rotate(300deg);
        -webkit-animation-timing-function: linear;
        -webkit-origin:0%;
    }

    30% {
        opacity: 1;
        -webkit-transform:rotate(410deg);
        -webkit-animation-timing-function: ease-in-out;
        -webkit-origin:7%;
    }

    39% {
        opacity: 1;
        -webkit-transform: rotate(645deg);
        -webkit-animation-timing-function: linear;
        -webkit-origin:30%;
    }

    70% {
        opacity: 1;
        -webkit-transform: rotate(770deg);
        -webkit-animation-timing-function: ease-out;
        -webkit-origin:39%;
    }

    75% {
        opacity: 1;
        -webkit-transform: rotate(900deg);
        -webkit-animation-timing-function: ease-out;
        -webkit-origin:70%;
    }

    76% {
    opacity: 0;
        -webkit-transform:rotate(900deg);
    }

    100% {
    opacity: 0;
        -webkit-transform: rotate(900deg);
    }
}

@-moz-keyframes orbit {
    0% {
        opacity: 1;
        z-index:99;
        -moz-transform: rotate(180deg);
        -moz-animation-timing-function: ease-out;
    }

    7% {
        opacity: 1;
        -moz-transform: rotate(300deg);
        -moz-animation-timing-function: linear;
        -moz-origin:0%;
    }

    30% {
        opacity: 1;
        -moz-transform:rotate(410deg);
        -moz-animation-timing-function: ease-in-out;
        -moz-origin:7%;
    }

    39% {
        opacity: 1;
        -moz-transform: rotate(645deg);
        -moz-animation-timing-function: linear;
        -moz-origin:30%;
    }

    70% {
        opacity: 1;
        -moz-transform: rotate(770deg);
        -moz-animation-timing-function: ease-out;
        -moz-origin:39%;
    }

    75% {
        opacity: 1;
        -moz-transform: rotate(900deg);
        -moz-animation-timing-function: ease-out;
        -moz-origin:70%;
    }

    76% {
    opacity: 0;
        -moz-transform:rotate(900deg);
    }

    100% {
    opacity: 0;
        -moz-transform: rotate(900deg);
    }
}
```

2. `Loading.js`

```
import React,{Component} from 'react';
import { connect } from 'dva';
import './loading.css';
class Loading extends Component {

    render() {
        const { isShow } = this.props;
        console.log(isShow)
        return (
            <div style={ {display: (isShow?'block':'none') } }>
                <div className="windows8"  >
                <div className="wBall" id="wBall_1">
                    <div className="wInnerBall"></div>
                </div>
                <div className="wBall" id="wBall_2">
                    <div className="wInnerBall"></div>
                </div>
                <div className="wBall" id="wBall_3">
                    <div className="wInnerBall"></div>
                </div>
                <div className="wBall" id="wBall_4">
                    <div className="wInnerBall"></div>
                </div>
                <div className="wBall" id="wBall_5">
                    <div className="wInnerBall"></div>
                </div>
            </div>
            </div>

        )
    }
}

export default connect(state=>{
    return {
        isShow:state.myloading.global
    }
})(Loading);
```

4. 在`src/routers/index` 中使用

```
// 引入
import Loading from '../components/common/Loading.js';
```

```

<Router history={history}>
                <div>
                    // 使用loading
                    <Loading/>
                    <Switch>
                       {/*默认访问signin*/}
                        <Route path="/home" component={Home}/>
                        <Route path="/signin" exact component={Signin}/>
                        <Redirect to="/signin"/>
                    </Switch>
                </div>

        </Router>
```