

讲师管理系统

1 开发流程

1.1 第一天

1. 初始化项目

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

2. 使用dva引入History和dva

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

3. 改造src/index.js通过dva的方式启动项目

```
1 // import React from 'react';
2 // import ReactDOM from 'react-dom';
3 // 使用history_1: 引入createHistory
4 import { createBrowserHistory } from 'history'
5
6 // 引入样式 使用 bootstrap
7 import 'bootstrap/dist/css/bootstrap.min.css'
8 import 'font-awesome/css/font-awesome.css'
9 import './style/index.css'
10
11 import './index.css';
12 // import App from './App';
13 import * as serviceWorker from './serviceWorker';
14
15 // 通过dva 的方式启动项目
16 // 1、引入DVA
17 import Dva from 'dva'
18 // 默认查找index首页
19 import router from './routers'
20 import model from './models'
21
22 // 2、创建dva的实例，APP对象
23 // 使用history_2: 传入Dva
24 const app = new Dva({
25   history: createBrowserHistory()
26 })
27
28 // 3、配置路由app.router(fn)
29 app.router(router)
30
31 // 4、app.use(); // 安装插件
32
33 // 5、注册模块 app.model
34 app.model(model)
35
```

```

36 // 5.5、卸载模块 app.unmodel()
37
38 // 6、app.start('#root') // root 就是根节点的id
39 app.start('#root')
40
41
42 serviceWorker.unregister();
43
44

```

4. 编写 src/router/index.js

```

1
2 // 配置路由规则
3 import React from 'react'
4 import { Router, Route, NavLink, Switch, Redirect } from 'dva/router'
5
6 const Home = () => <h1>您好</h1>
7
8 // 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】
9 // history 没有 # 号
10 let fn = function ({ history, app }) {
11   return (
12     <Router history={ history }>
13       <React.Fragment>
14         // exact 不做嵌套路由
15         <Route path="/home" exact component={ Home } />
16       </React.Fragment>
17     </Router>
18   )
19 }
20
21
22 // 导出为函数
23 export default fn;

```

5. 编写 src/model/index.js

```

1 let index = {
2   // 命名空间 具体命名
3   namespace: 'index',
4   state: {
5     num: 0
6   },
7   effects: {
8     // 异步action
9     /* payload: 发来的action 中携带的数据
10    select: 获取当前的state
11    put: 调用reducers
12    call: 调用异步方法 */
13    *changeNum ({ payload }, { select, put, call }) {
14      // 保持reducers 中名字的一致性
15      console.log(payload)
16      yield put({ type: 'addNum', payload })
17    }

```

```

18   },
19   reducers: {
20     /*
21     let two = {
22       payload: {num: 8}
23       type: "index/addNum"
24     }
25     let { payload } = two // { num: 8 }
26     */
27     addNum (state, { payload }) {
28       // 返回一个新对象 值不可变性
29       console.log(state, payload)
30       return {
31         num: state.num + payload.num
32       }
33     }
34   }
35 }
36
37
38 export default index

```

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

```

1  import React, { Component } from 'react'
2  // 使用connect_1 引入
3  import { connect } from 'dva'
4
5  class Home extends Component {
6    render () {
7      console.log(this.props.num)
8      return (
9        <React.Fragment>
10         <h1>
11           数量: { this.props.num }
12         </h1>
13         <button onClick={ e=> {
14           // 带上命名空间 , 并传递payload
15           this.props.dispatch({ type: 'index/changeNum', payload: {
16             num:8 } })
17         } }>
18           增加number
19         </button>
20       </React.Fragment>
21     )
22   }
23
24   export default connect(state => {
25     console.log(state)
26     // 使用connect_2 声明在prop中需要使用的
27     return {
28       // 通过命名去state 中去取
29       num: state.index.num
30     }
31   })(Home)
32

```

7. 使用 axios

```
1 | npm i axios -S
```

1. 封装请求

src/config.js

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

2. src/utils/request.js

```
1 | import Axios from 'axios'
2 | import { serverHost, port } from '../config/index'
3 |
4 | let r = Axios.create({
5 |   baseURL: `${serverHost}:${port}/`
6 | })
7 |
8 | let request = function (url='', options={}) {
9 |   if (url === '') {
10 |     return Promise.reject('必须传递url')
11 |   }
12 |   return r({
13 |     url,
14 |     method: 'get',
15 |     ...options
16 |   })
17 | }
18 |
19 | export default request
```

1.2 登录

1. src/components/Signin.js

```
1 | import React, { Component } from 'react'
2 |
3 | // 处理图片 手动告知webpack 加载图片
4 | import monkey from '../assets/img/monkey.png'
5 |
6 | class Signin extends Component {
7 |   constructor () {
8 |     super()
9 |     this.state = {
10 |       username: '',
11 |       password: ''
12 |     }
13 |   }
14 |
15 |   // 调用teacher中的登录异步方法
16 |   handelSignin (e) {
17 |     e.preventDefault()
18 |     let { username, password } = this.state
```

```

19     console.log(username)
20     this.props.dispatch({
21       type: 'teacher/doLogin',
22       payload: {
23         username,
24         password
25       }
26     })
27   }
28   render () {
29     return (
30       <React.Fragment>
31         <form action="" className="col-xs-offset-1 col-xs-10"
onSubmit={ e=> this.handleSignin() }>
32           <input id="name" type="text" className="form-control"
placeholder="用户名"
33             value={ this.state.username } onChange={e=> {
34               this.setState({
35                 username: e.target.value
36               })
37             }}
38           />
39           <input id="pass" type="password" className="form-
control" placeholder="密码"
40             value={ this.state.password } onChange={e=> {
41               this.setState({
42                 password: e.target.value
43               })
44             }}
45           />
46         </React.Fragment>
47       </React.Fragment>
48     )
49   }
50 }
51
52
53 export default connect(state=> {
54   return {
55     isLogin: state.teacher.isLogin
56   }
57 })(Signin)

```

2. src/routers/index.js

在路由中注册

```

1  import Signin from '../components/Signin'
2
3  // 使用history 需要先进行声明
4  let fn = function ({ history, app }) {
5    return (
6      <Router history={ history }>
7        <React.Fragment>
8          <Route path="/home" exact component={ Home } />
9          <Route path="/signin" exact component={ Signin } />
10         </React.Fragment>

```

```

11     </Router>
12   )
13 }
14
15
16 // 导出为函数
17 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

```

1  import request from '../utils/request'
2  // 做路由的跳转使用
3  import { routerRedux } from 'dva/router'
4
5  import * as api from '../api/index'
6
7  let teacher = {
8    // 命名空间 具体命名
9    namespace: 'teacher',
10    state: {
11      isLogin: false
12    },
13    effects: {
14      // 异步action 同步返回 结果
15      /* payload: 发来的action 中携带的数据
16       select: 获取当前的state
17       put: 调用reducers
18       call: 调用异步方法 */
19      *doLogin ({ payload }, { select, put, call }) {
20        let res
21        try {
22          // 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
23          Promise
24          // 同步返回
25          /*
26             let request = function (url='', options={}) {
27               return function () {
28                 if (url === '') {
29                   return Promise.reject('必须传递url')
30                 }
31                 return r({
32                   url,
33                   method: 'get',
34                   ...options
35                 })
36               }
37             }
38             res = yield call(request('signin', payload))
39             */
40            res = yield call(() => api.login(payload))
41          } catch (error) {
42            console.log('登录失败', error)
43          }
44          if (res.data.code === 0) {
45            console.log(res.data)

```

```

45     window.sessionStorage.setItem('user',
JSON.stringify(res.data.data.user))
46     yield put({type: 'changeLogin', payload: {isLogin: true}})
47     //
48     yield put(routerRedux.push('/home'))
49   }
50 }
51 },
52 reducers: {
53   /*
54     整个对象是 action 只要action中的payload
55     action: {
56       payload: {num: 8}
57       type: "index/addNum"
58     }
59     let { payload } = two // { num: 8 }
60   */
61   changeLogin (state, { payload }) {
62     return {
63       isLogin: payload.isLogin
64     }
65   }
66 }
67 }
68
69
70 export default teacher

```

基于 action 进行页面跳转

在effects中进行路由的跳转

```

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

```

1 // 注册teacher model
2 import teacher from './models/teacher'
3
4 // 注册 teacher 模块
5 app.model(teacher)

```

1.3 首页

1.3.1 重定向到登录页

src/routers/index

```
1   <Router history={ history }>
2     <React.Fragment>
3       <Switch>
4         <Route path="/home" component={ Home } />
5         <Route path="/signin" exact component={ Signin } />
6         // 重定向 外层需要有Switch 做横向的匹配
7         <Redirect to="/signin" />
8       </Switch>
9     </React.Fragment>
10  </Router>
```

1.3.2 首页嵌套路由

src/routers/index

```
1  export const MyRoute = Route;
2  // 首页的嵌套路由表
3  export const homeSubRouters = [
4    {
5      path: '/home/list',
6      component: TeacherList
7    }
8  ]
9
10 ...
11 // 嵌套路由不能有exact(精确匹配) 关键字
12 // 因为如果做精确匹配的话 父组件就显示不出来
13 <Route path="/home" component={ Home } />
14 ...
```

src/components/Home.js

```
1  import { homeSubRouters, MyRoute as Route, MyRouter as Router } from
2    '../routers';
3
4  { /* 嵌套路由 */
5    {
6      homeSubRouters.map((route, i) => {
7        return <Route key={i} path={route.path} component=
8          {route.component} />
9      })
10    }
11  }
```

1.4 使用Loading

1.4.1 全局Loading

1. 安装 dva-loading

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

2. 安装插件 src/index.js


```

1 import createLoading from 'dva-loading';
2
3 // 4、app.use(); // 安装插件
4 app.use(createLoading({
5   // 创建了一个命名空间
6   namespace: 'myloading', //默认是loading没有啥用
7 }));

```

3. 创建 Loading.css 和 Loading.js

1. Loading.css

```

1 .windows8 {
2   position: relative;
3   width: 78px;
4   height: 78px;
5   margin: auto;
6 }
7
8 .windows8 .wBall {
9   position: absolute;
10  width: 74px;
11  height: 74px;
12  opacity: 0;
13  transform: rotate(225deg);
14  -o-transform: rotate(225deg);
15  -ms-transform: rotate(225deg);
16  -webkit-transform: rotate(225deg);
17  -moz-transform: rotate(225deg);
18  animation: orbit 6.96s infinite;
19  -o-animation: orbit 6.96s infinite;
20  -ms-animation: orbit 6.96s infinite;
21  -webkit-animation: orbit 6.96s infinite;
22  -moz-animation: orbit 6.96s infinite;
23 }
24
25 .windows8 .wBall .winnerBall{
26   position: absolute;
27   width: 10px;
28   height: 10px;
29   background: rgb(0,0,0);
30   left: 0px;
31   top: 0px;
32   border-radius: 10px;
33 }
34
35 .windows8 #wBall_1 {
36   animation-delay: 1.52s;
37   -o-animation-delay: 1.52s;
38   -ms-animation-delay: 1.52s;
39   -webkit-animation-delay: 1.52s;
40   -moz-animation-delay: 1.52s;
41 }
42
43 .windows8 #wBall_2 {
44   animation-delay: 0.3s;
45   -o-animation-delay: 0.3s;

```

```
46     -ms-animation-delay: 0.3s;
47     -webkit-animation-delay: 0.3s;
48     -moz-animation-delay: 0.3s;
49 }
50
51 .windows8 #wBall_3 {
52     animation-delay: 0.61s;
53     -o-animation-delay: 0.61s;
54     -ms-animation-delay: 0.61s;
55     -webkit-animation-delay: 0.61s;
56     -moz-animation-delay: 0.61s;
57 }
58
59 .windows8 #wBall_4 {
60     animation-delay: 0.91s;
61     -o-animation-delay: 0.91s;
62     -ms-animation-delay: 0.91s;
63     -webkit-animation-delay: 0.91s;
64     -moz-animation-delay: 0.91s;
65 }
66
67 .windows8 #wBall_5 {
68     animation-delay: 1.22s;
69     -o-animation-delay: 1.22s;
70     -ms-animation-delay: 1.22s;
71     -webkit-animation-delay: 1.22s;
72     -moz-animation-delay: 1.22s;
73 }
74
75
76
77 @keyframes orbit {
78     0% {
79         opacity: 1;
80         z-index:99;
81         transform: rotate(180deg);
82         animation-timing-function: ease-out;
83     }
84
85     7% {
86         opacity: 1;
87         transform: rotate(300deg);
88         animation-timing-function: linear;
89         origin:0%;
90     }
91
92     30% {
93         opacity: 1;
94         transform:rotate(410deg);
95         animation-timing-function: ease-in-out;
96         origin:7%;
97     }
98
99     39% {
100         opacity: 1;
101         transform: rotate(645deg);
102         animation-timing-function: linear;
103         origin:30%;
```

```
104     }
105
106     70% {
107         opacity: 1;
108         transform: rotate(770deg);
109         animation-timing-function: ease-out;
110         origin: 39%;
111     }
112
113     75% {
114         opacity: 1;
115         transform: rotate(900deg);
116         animation-timing-function: ease-out;
117         origin: 70%;
118     }
119
120     76% {
121         opacity: 0;
122         transform: rotate(900deg);
123     }
124
125     100% {
126         opacity: 0;
127         transform: rotate(900deg);
128     }
129 }
130
131 @-o-keyframes orbit {
132     0% {
133         opacity: 1;
134         z-index: 99;
135         -o-transform: rotate(180deg);
136         -o-animation-timing-function: ease-out;
137     }
138
139     7% {
140         opacity: 1;
141         -o-transform: rotate(300deg);
142         -o-animation-timing-function: linear;
143         -o-origin: 0%;
144     }
145
146     30% {
147         opacity: 1;
148         -o-transform: rotate(410deg);
149         -o-animation-timing-function: ease-in-out;
150         -o-origin: 7%;
151     }
152
153     39% {
154         opacity: 1;
155         -o-transform: rotate(645deg);
156         -o-animation-timing-function: linear;
157         -o-origin: 30%;
158     }
159
160     70% {
161         opacity: 1;
```

```
162     -o-transform: rotate(770deg);
163     -o-animation-timing-function: ease-out;
164     -o-origin: 39%;
165 }
166
167 75% {
168     opacity: 1;
169     -o-transform: rotate(900deg);
170     -o-animation-timing-function: ease-out;
171     -o-origin: 70%;
172 }
173
174 76% {
175     opacity: 0;
176     -o-transform: rotate(900deg);
177 }
178
179 100% {
180     opacity: 0;
181     -o-transform: rotate(900deg);
182 }
183 }
184
185 @-ms-keyframes orbit {
186     0% {
187         opacity: 1;
188         z-index: 99;
189         -ms-transform: rotate(180deg);
190         -ms-animation-timing-function: ease-out;
191     }
192
193     7% {
194         opacity: 1;
195         -ms-transform: rotate(300deg);
196         -ms-animation-timing-function: linear;
197         -ms-origin: 0%;
198     }
199
200     30% {
201         opacity: 1;
202         -ms-transform: rotate(410deg);
203         -ms-animation-timing-function: ease-in-out;
204         -ms-origin: 7%;
205     }
206
207     39% {
208         opacity: 1;
209         -ms-transform: rotate(645deg);
210         -ms-animation-timing-function: linear;
211         -ms-origin: 30%;
212     }
213
214     70% {
215         opacity: 1;
216         -ms-transform: rotate(770deg);
217         -ms-animation-timing-function: ease-out;
218         -ms-origin: 39%;
219     }
```

```
220
221     75% {
222         opacity: 1;
223         -ms-transform: rotate(900deg);
224         -ms-animation-timing-function: ease-out;
225         -ms-origin: 70%;
226     }
227
228     76% {
229         opacity: 0;
230         -ms-transform: rotate(900deg);
231     }
232
233     100% {
234         opacity: 0;
235         -ms-transform: rotate(900deg);
236     }
237 }
238
239 @-webkit-keyframes orbit {
240     0% {
241         opacity: 1;
242         z-index: 99;
243         -webkit-transform: rotate(180deg);
244         -webkit-animation-timing-function: ease-out;
245     }
246
247     7% {
248         opacity: 1;
249         -webkit-transform: rotate(300deg);
250         -webkit-animation-timing-function: linear;
251         -webkit-origin: 0%;
252     }
253
254     30% {
255         opacity: 1;
256         -webkit-transform: rotate(410deg);
257         -webkit-animation-timing-function: ease-in-out;
258         -webkit-origin: 7%;
259     }
260
261     39% {
262         opacity: 1;
263         -webkit-transform: rotate(645deg);
264         -webkit-animation-timing-function: linear;
265         -webkit-origin: 30%;
266     }
267
268     70% {
269         opacity: 1;
270         -webkit-transform: rotate(770deg);
271         -webkit-animation-timing-function: ease-out;
272         -webkit-origin: 39%;
273     }
274
275     75% {
276         opacity: 1;
277         -webkit-transform: rotate(900deg);
```

```
278     -webkit-animation-timing-function: ease-out;
279     -webkit-origin: 70%;
280 }
281
282 76% {
283     opacity: 0;
284     -webkit-transform: rotate(900deg);
285 }
286
287 100% {
288     opacity: 0;
289     -webkit-transform: rotate(900deg);
290 }
291 }
292
293 @-moz-keyframes orbit {
294     0% {
295         opacity: 1;
296         z-index: 99;
297         -moz-transform: rotate(180deg);
298         -moz-animation-timing-function: ease-out;
299     }
300
301     7% {
302         opacity: 1;
303         -moz-transform: rotate(300deg);
304         -moz-animation-timing-function: linear;
305         -moz-origin: 0%;
306     }
307
308     30% {
309         opacity: 1;
310         -moz-transform: rotate(410deg);
311         -moz-animation-timing-function: ease-in-out;
312         -moz-origin: 7%;
313     }
314
315     39% {
316         opacity: 1;
317         -moz-transform: rotate(645deg);
318         -moz-animation-timing-function: linear;
319         -moz-origin: 30%;
320     }
321
322     70% {
323         opacity: 1;
324         -moz-transform: rotate(770deg);
325         -moz-animation-timing-function: ease-out;
326         -moz-origin: 39%;
327     }
328
329     75% {
330         opacity: 1;
331         -moz-transform: rotate(900deg);
332         -moz-animation-timing-function: ease-out;
333         -moz-origin: 70%;
334     }
335 }
```

```

336 76% {
337  opacity: 0;
338      -moz-transform: rotate(900deg);
339  }
340
341 100% {
342  opacity: 0;
343      -moz-transform: rotate(900deg);
344  }
345  }

```

2. Loading.js

```

1  import React,{Component} from 'react';
2  import { connect } from 'dva';
3  import './loading.css';
4  class Loading extends Component {
5
6      render() {
7          const { isShow } = this.props;
8          console.log(isShow)
9          return (
10             <div style={ {display: (isShow?'block':'none') } }>
11                 <div className="windows8" >
12                     <div className="wBall" id="wBall_1">
13                         <div className="winnerBall"></div>
14                     </div>
15                     <div className="wBall" id="wBall_2">
16                         <div className="winnerBall"></div>
17                     </div>
18                     <div className="wBall" id="wBall_3">
19                         <div className="winnerBall"></div>
20                     </div>
21                     <div className="wBall" id="wBall_4">
22                         <div className="winnerBall"></div>
23                     </div>
24                     <div className="wBall" id="wBall_5">
25                         <div className="winnerBall"></div>
26                     </div>
27                 </div>
28             </div>
29         )
30     }
31 }
32
33
34 export default connect(state=>{
35     return {
36         isShow:state.myloading.global
37     }
38 })(Loading);

```

4. 在 src/routers/index 中使用

```

1  // 引入
2  import Loading from '../components/common/Loading.js';

```

```
3
4 <Router history={history}>
5     <div>
6         // 使用loading
7         <Loading/>
8         <Switch>
9             { /*默认访问signin*/ }
10            <Route path="/home" component={Home}/>
11            <Route path="/signin" exact component=
{Signin}/>
12                <Redirect to="/signin"/>
13            </Switch>
14        </div>
15
16    </Router>
```