## **1.搭建开发环境**

### **1.1 package.json**

{

"name": "zhufeng-ts-ketang",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"dev": "webpack-dev-server"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"@types/react": "^16.8.3",

"@types/react-dom": "^16.8.1",

"@types/react-redux": "^7.0.1",

"@types/react-router-dom": "^4.3.1",

"@types/redux-logger": "^3.0.7",

"connected-react-router": "^6.3.1",

"react": "^16.8.2",

"react-dom": "^16.8.2",

"react-redux": "^6.0.0",

"react-router-dom": "^4.3.1",

"redux": "^4.0.1",

"redux-logger": "^3.0.6",

"redux-thunk": "^2.3.0"

},

"devDependencies": {

"html-webpack-plugin": "^3.2.0",

"source-map-loader": "^0.2.4",

"ts-loader": "^5.3.3",

"typescript": "^3.3.3",

"webpack": "^4.29.3",

"webpack-cli": "^3.2.3",

"webpack-dev-server": "^3.1.14"

}

}

### **1.2 tsconfig.json**

{

"compilerOptions": {

"outDir": "./dist",

"sourceMap": true,

"noImplicitAny": true,

"module": "commonjs",

"target": "es5",

"jsx": "react"

}

}

### **1.3 webpack.config.js**

**const** webpack=require('webpack');**const** HtmlWebpackPlugin=require('html-webpack-plugin');**const** path=require('path');module.exports={

mode: 'development',

entry: "./src/index.tsx",

output: {

filename: "bundle.js",

path: path.join(\_\_dirname,'dist')

},

devtool: "source-map",

devServer: {

hot: true,

contentBase: path.join(\_\_dirname,'dist'),

historyApiFallback: {

index:'./index.html'

}

},

resolve: {

extensions: [".ts", ".tsx", ".js", ".json"]

},

module: {

rules: [{

test: /\.tsx?$/,

loader: "ts-loader"

},

{

enforce: "pre",

test: /\.js$/,

loader: "source-map-loader"

}

]

},

plugins: [

**new** HtmlWebpackPlugin({

template:'./src/index.html'

}),

**new** webpack.HotModuleReplacementPlugin()

],

};

### **1.4 src\index.tsx**

src\index.tsx

**import** \* **as** React **from** 'react';**import** \* **as** ReactDOM **from** 'react-dom';**import** Counter1 **from** './components/Counter1';**import** Counter2 **from** './components/Counter2';**import** { Provider } **from** 'react-redux';**import** store **from** './store';**import** {Route,Link } **from** 'react-router-dom';**import** { ConnectedRouter } **from** 'connected-react-router'**import** history **from** './store/history';

ReactDOM.render((

<Provider store={store}>

<ConnectedRouter history={history}>

<React.Fragment>

<Link to="/counter1">counter1</Link>

<Link to="/counter2">counter2</Link>

<Route path="/counter1" component={Counter1} />

<Route path="/counter2" component={Counter2}/>

</React.Fragment>

</ConnectedRouter>

</Provider>

),document.getElementById('root'));

### **1.5 src\index.html**

src\index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="http://at.alicdn.com/t/font\_pgg5jafnob51m7vi.css">

<title>珠峰课堂</title>

</head>

<body>

<div id="root"></div>

</body>

</html>

### **1.6 components\Counter1.tsx**

src\components\Counter1.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** \* **as** types **from** '../store/types';**import** \* **as** actions **from** '../store/actions/counter1';**export** interface Props{

number: number,

increment1: any,

decrement1: any,

goCounter2: any

}**class** **Counter1** **extends** **React**.**Component**<**Props**>{

render() {

**const** {number,increment1,decrement1,goCounter2}=**this**.props;

**return** (

<div>

<p>{number}</p>

<button onClick={increment1}>+</button>

<button onClick={decrement1}>-</button>

<button onClick={goCounter2}>goCounter2</button>

</div>

)

}

}

**let** mapStateToProps=**function** (state:types.Store):**types**.**Counter1** {

**return** state.counter1;

}

**export** **default** connect(mapStateToProps,actions)(Counter1);

### **1.7 components\Counter2.tsx**

src\components\Counter2.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** \* **as** types **from** '../store/types';**import** \* **as** actions **from** '../store/actions/counter2';**export** interface Props{

number: number,

increment2: any,

decrement2: any

}**class** **Counter2** **extends** **React**.**Component**<**Props**>{

render() {

**const** {number,increment2,decrement2}=**this**.props;

**return** (

<div>

<p>{number}</p>

<button onClick={increment2}>+</button>

<button onClick={decrement2}>-</button>

</div>

)

}

}

**let** mapStateToProps=**function** (state:types.Store):**types**.**Counter2** {

**return** state.counter2;

}

**export** **default** connect(mapStateToProps,actions)(Counter2);

### **1.8 store\index.tsx**

src\store\index.tsx

**import** {createStore,applyMiddleware} **from** 'redux'**import** reducers **from** './reducers';**import** { routerMiddleware } **from** 'connected-react-router'**import** history **from** './history';**import** thunk **from** 'redux-thunk';**import** logger **from** 'redux-logger';**let** router = routerMiddleware(history);**let** store=createStore(reducers,applyMiddleware(router,thunk,logger));**export** **default** store;

### **1.9 history.tsx**

src\store\history.tsx

**import** {createBrowserHistory} **from** 'history'**const** history=createBrowserHistory()**export** **default** history;

### **1.10 store\action-types.tsx**

src\store\action-types.tsx

**export** **const** INCREMENT='INCREMENT';**export** **const** DECREMENT='DECREMENT';

**export** **const** INCREMENT1='INCREMENT1';**export** **const** DECREMENT1='DECREMENT1';

**export** **const** INCREMENT2='INCREMENT2';**export** **const** DECREMENT2='DECREMENT2';

### **1.11 store\types\index.tsx**

src\store\types\index.tsx

**export** interface Store{

counter1: Counter1,

counter2: Counter2

}**export** interface Counter1{

number: number

}**export** interface Counter2{

number: number

}

### **1.12 store\reducers\index.tsx**

src\store\reducers\index.tsx

**import** counter1 **from** './counter1';**import** counter2 **from** './counter2';**import** { combineReducers } **from** 'redux';**import** history **from** '../history';**import** { connectRouter } **from** 'connected-react-router'**let** reducers=combineReducers({

counter1,

counter2,

router: connectRouter(history)

});**export** **default** reducers;

### **1.13 store\reducers\counter1.tsx**

src\store\reducers\counter1.tsx

**import** \* **as** types **from** '../action-types';**import** { Counter1 } **from** '../types';**import** {Action} **from** '../actions/counter1';**export** **default** **function** (state: Counter1={ number: 0 },action: Action): **Counter1** {

**switch** (action.type) {

**case** types.INCREMENT1:

**return** {...state,number:state.number+1};

**case** types.DECREMENT1:

**return** {...state,number:state.number-1};

**default**:

**return** state;

}

}

### **1.14 store\reducers\counter2.tsx**

src\store\reducers\counter2.tsx

**import** \* **as** types **from** '../action-types';**import** { Counter2 } **from** '../types';**import** {Action} **from** '../actions/counter2';**export** **default** **function** (state: Counter2={ number: 0 },action: Action): **Counter2** {

**switch** (action.type) {

**case** types.INCREMENT2:

**return** {...state,number:state.number+1};

**case** types.DECREMENT2:

**return** {...state,number:state.number-1};

**default**:

**return** state;

}

}

### **1.15 store\actions\counter1.tsx**

src\store\actions\counter1.tsx

**import** {INCREMENT1,DECREMENT1} **from** '../action-types';**import** { push } **from** 'connected-react-router';

**export** interface Increment1{

type:**typeof** INCREMENT1

}**export** interface Decrement1{

type:**typeof** DECREMENT1

}**export** type Action=Increment1|Decrement1;

**export** **function** **increment1**(): **any** {

**return** **function** (dispatch:any,getState:any) {

setTimeout(**function** () {

dispatch({

type:INCREMENT1

})

},1000);

}

}**export** **function** **decrement1**():**Decrement1** {

**return** { type: DECREMENT1 };

}**export** **function** **goCounter2**():**any** {

**return** push('/counter2');

}

### **1.16 store\actions\counter2.tsx**

src\store\actions\counter2.tsx

**import** {INCREMENT2,DECREMENT2} **from** '../action-types';**export** interface Increment2{

type:**typeof** INCREMENT2

}**export** interface Decrement2{

type:**typeof** DECREMENT2

}**export** type Action=Increment2|Decrement2;

**export** **function** **increment2**(): **Increment2** {

**return** { type: INCREMENT2 };

}**export** **function** **decrement2**():**Decrement2** {

**return** { type: DECREMENT2 };

}

## **2. 项目准备**

* [PS素材](http://img.zhufengpeixun.cn/zhufengketang-resource.zip)
* [图标链接](http://at.alicdn.com/t/font_pgg5jafnob51m7vi.css)
  + .icon-uilist
  + .icon-xiaolian
  + .icon-book
  + .icon-fanhui
  + .icon-guanbi
  + .icon-xingqiu
  + .icon-kecheng-copy
  + .icon-react



## **3. 首页导航**

### **3.1 package.json**

"devDependencies": {+ "css-loader": "^2.1.0",+ "less": "^3.9.0",+ "less-loader": "^4.1.0",+ "style-loader": "^0.23.1",

}

}

### **3.2 webpack.config.js**

webpack.config.js

devServer: {

hot: true,

contentBase: path.join(\_\_dirname,'dist'),+ historyApiFallback: true

},+ {+ test: /\.less$/,+ use:['style-loader','css-loader','less-loader']+ }

### **3.3 src/index.tsx**

src/index.tsx

**import** \* **as** React **from** 'react';**import** \* **as** ReactDOM **from** 'react-dom';**import** { Provider } **from** 'react-redux';**import** store **from** './store';**import** {Route,Link } **from** 'react-router-dom';**import** { ConnectedRouter } **from** 'connected-react-router'**import** history **from** './store/history';**import** App **from** './containers/App';

ReactDOM.render((

<Provider store={store}>

<ConnectedRouter history={history}>

<Route component={App}/>

</ConnectedRouter>

</Provider>

),document.getElementById('root'));

### **3.4 common\index.less**

src\common\index.less

\*{

margin: 0;

padding: 0;

}

ul,li{

list-style: none;

}

a{

text-decoration: none;

}

html,body,#root{

width:100%;

height:100%;

overflow: hidden;

}

### **3.5 components\Tab\index.tsx**

src\components\Tab\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** {NavLink} **from** 'react-router-dom';**import** './index.less';

interface Props{

}**class** **Tab** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<nav className="footer">

<NavLink exact to="/" activeClassName="active">

<i className="iconfont icon-xingqiu"></i>

首页

</NavLink>

<NavLink to="/mime" activeClassName="active">

<i className="iconfont icon-react"></i>

我的课程

</NavLink>

<NavLink to="/profile" activeClassName="active">

<i className="iconfont icon-xiaolian"></i>

个人中心

</NavLink>

</nav>

)

}

}**export** **default** connect()(Tab);

### **3.6 components\Tab\index.less**

src\components\Tab\index.less

.footer{

position: fixed;

width:100%;

height:53px;

bottom:0;

display: flex;

background-color: #FFF;

border-top:1px solid #d5d5d5;

a{

flex:1;

display: flex;

color:#b5b5b6;

flex-direction: column;

justify-content: center;

align-items: center;

&.active{

color:#188ae4;

}

}

}

### **3.6 containers\App.tsx**

src\containers\App.tsx

**import** {Route,Link,Switch} **from** 'react-router-dom';**import** \* **as** React **from** 'react';**import** Tab **from** '../components/Tab';**import** Home **from** './Home';**import** Mime **from** './Mime';**import** Profile **from** './Profile';**import** '../common/index.less';

interface IProps{

children:any

}**export** **default** **class** **App** **extends** **React**.**Component**<**IProps**>{

render(){

**return** (

<React.Fragment>

<Route exact path="/" component={Home}/>

<Route path="/mime" component={Mime}/>

<Route path="/profile" component={Profile}/>

<Tab></Tab>

</React.Fragment>

)

}

}

### **3.7 containers\Home\index.tsx**

src\containers\Home\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';

interface Props{

}**class** **Home** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<div >

Home

</div>

)

}

}**export** **default** connect()(Home);

### **3.8 containers\Mime\index.tsx**

src\containers\Mime\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';

interface Props{

}**class** **Mime** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<div >

Mime

</div>

)

}

}**export** **default** connect()(Mime);

### **3.9 containers\Profile\index.tsx**

src\containers\Profile\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';

interface Props{

}**class** **Profile** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<div >

Profile

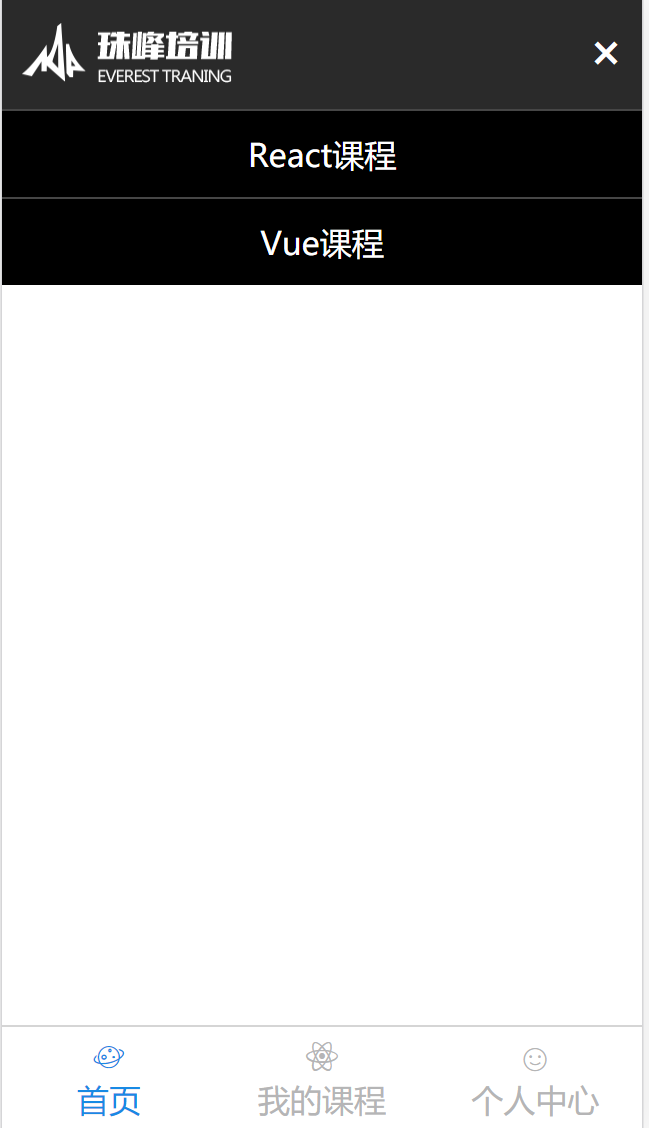
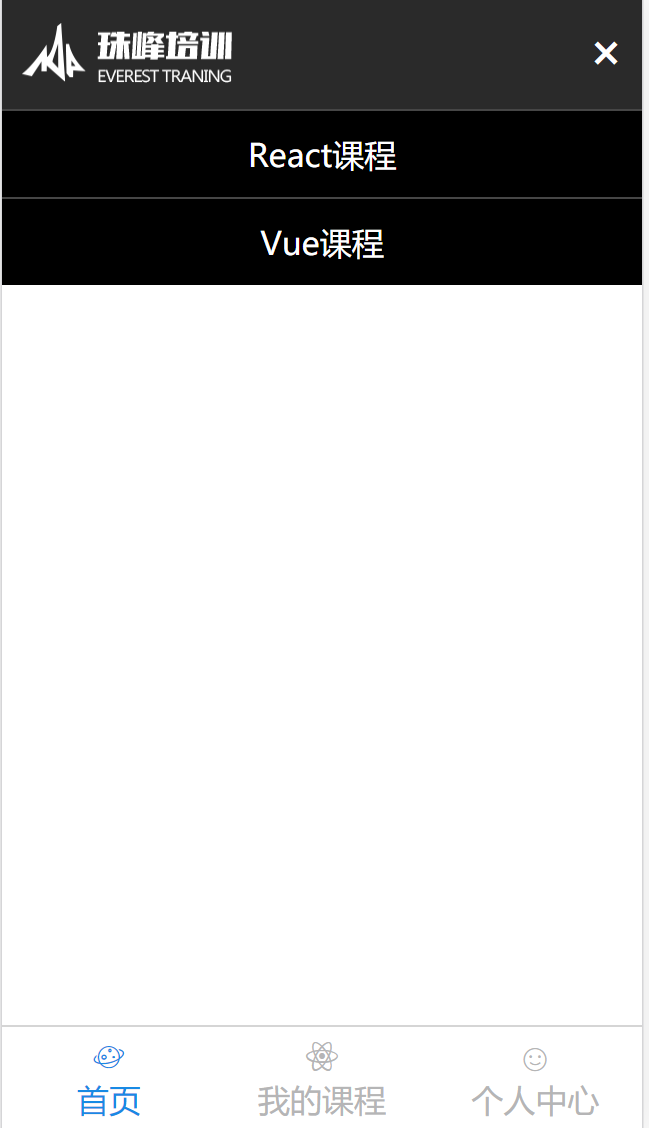
</div>

)

}

}**export** **default** connect()(Profile);

## **4. 首页头部动画**

### **4.1 containers/Home/index.tsx**

src/containers/Home/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** HomeHeader **from** './components/Header';**import** './index.less';

interface Props{}**class** **Home** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<React.Fragment>

<HomeHeader/>

</React.Fragment>

)

}

}**export** **default** connect()(Home);

### **4.2 webpack.config.js**

{

test: /\.(jpg|png|gif)$/,

use:'url-loader'

}

### **4.3 Home\components\Header\index.tsx**

src\containers\Home\components\Header\index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** { CSSTransition,TransitionGroup } **from** 'react-transition-group';

declare **function** **require**(url: string): **string**;**const** **logo**=**require**('../../../../images/logo.png');**import** { Props } **from** '../../../../types';**export** **default** **class** **HomeHeader** **extends** **React**.**Component**<**Props**> {

state={

showList:false

}

render() {

**return** (

<div className="home-header">

<div className="header-menu">

<img src={logo} alt="logo" />

<div onClick={() => this.setState({showList:!this.state.showList})}>

{

this.state.showList?<i className="iconfont icon-guanbi"></i>:<i className="iconfont icon-uilist"></i>

}

</div>

</div>

<TransitionGroup>

{

this.state.showList&&<CSSTransition

timeout={500}

classNames="fade"

><ul className="menu-list">

<li data-category="1">React课程</li>

<li data-category="2">Vue课程</li>

</ul></CSSTransition>

}

</TransitionGroup>

</div>

);

}

}

### **4.4 Home\components\Header\index.less**

src\containers\Home\components\Header\index.less

.home-header{

height:56px;

background-color:#2a2a2a ;

width:100%;

position: fixed;

top:0;

left:0;

z-index: 10;

.header-menu{

height:56px;

display: flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

img{

width:105px;

height:30px;

margin-left:10px;

}

i{

color:#FFF;

margin-right:10px;

}

}

.menu-list{

position: absolute;

top:56px;

left:0;

width:100%;

background-color: #000;

li{

width:100%;

height:43px;

line-height: 43px;

border-top:1px solid #464646;

color:#FFF;

text-align: center;

&.active{

color:red;

}

}

}

}.fade-enter {

opacity: 0.01;

}.fade-enter-active {

opacity: 1;

transition: opacity 500ms ease-in;

}.fade-exit {

opacity: 1;

}.fade-exit-active {

opacity: 0.01;

transition: opacity 500ms ease-in;

}

### **4.5 types\index.tsx**

src\types\index.tsx

**export** interface Store{

counter1: Counter1,

counter2: Counter2

}**export** interface Counter1{

number: number

}**export** interface Counter2{

number: number

}**export** interface Props{

children?: any,

}

## **5. 当前分类存入redux**

### **5.1 containers/Home/index.tsx**

src/containers/Home/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** Header **from** './components/Header';**import** {Store} **from** '../../types';**import** actions **from** '../../store/actions/home';**import** './index.less';

interface Props{

category:string,

changeCategory:any

}**class** **Home** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<React.Fragment>

<Header

category={this.props.category}

changeCategory={this.props.changeCategory}

/>

</React.Fragment>

)

}

}**export** **default** connect(

(state:Store)=>state.home,

actions

)(Home);

### **5.2 store/action-types.tsx**

src/store/action-types.tsx

//改变当前的分类**export** **const** CHANGE\_CATEGORY = 'CHANGE\_CATEGORY';

### **5.3 store/action-types.tsx**

src/store/action-types.tsx

//改变当前的分类**export** **const** CHANGE\_CATEGORY = 'CHANGE\_CATEGORY';

### **5.4 store/reducers/index.tsx**

src/store/reducers/index.tsx

**import** { combineReducers } **from** 'redux';**import** history **from** '../history';**import** home **from** './home';**import** { connectRouter } **from** 'connected-react-router'**let** reducers=combineReducers({

router: connectRouter(history),

home

});**export** **default** reducers;

### **5.5 src\types\index.tsx**

src\types\index.tsx

**export** interface Store{

home:Home,

router:any

}**export** interface Home{

category:string,

}**export** interface Props{

children?: any,

}

### **5.6 store\actions\home.tsx**

src\store\actions\home.tsx

**import** \* **as** types **from** '../action-types';**export** interface changeCategory {

type:string,//改变当前的分类

payload:any //新的分类的名称

}//type是用来给类型起别名的**export** type Action = changeCategory;**export** **default** {

changeCategory(category:string):changeCategory{

**return** {type:types.CHANGE\_CATEGORY,payload:category};

}

}

### **5.7 store/actions/home.tsx**

src/store/actions/home.tsx

**import** \* **as** types **from** '../action-types';**export** interface changeCategory {

type:string,//改变当前的分类

payload:any //新的分类的名称

}//type是用来给类型起别名的**export** type Action = changeCategory;**export** **default** {

changeCategory(category:string):changeCategory{

**return** {type:types.CHANGE\_CATEGORY,payload:category};

}

}

### **5.8 store/reducers/home.tsx**

src/store/reducers/home.tsx

**import** {Home} **from** '../../types';**import** {Action} **from** '../actions/home';**import** \* **as** types **from** '../action-types';**let** initState:Home = {

category:'all'

};**export** **default** **function**(state:Home=initState,action:Action){

**switch**(action.type){

**case** types.CHANGE\_CATEGORY:

**return** {...state,category:action.payload};

**default**:

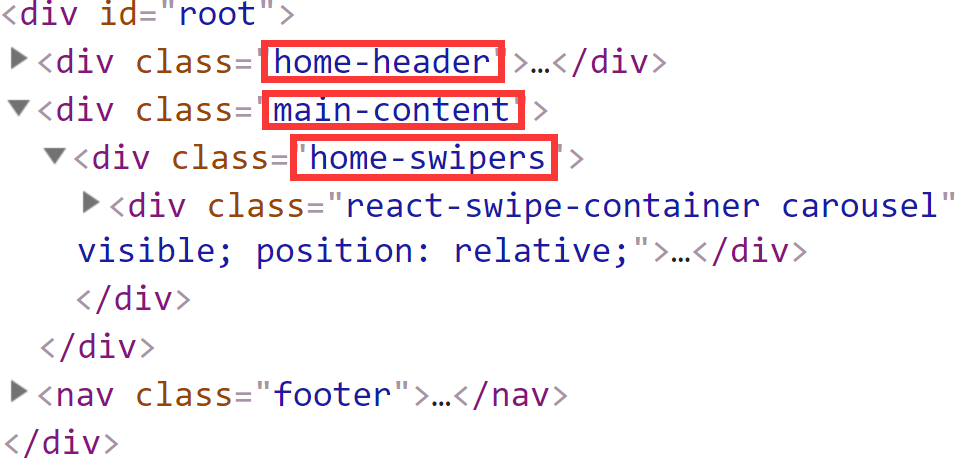
**return** state;

}

}

## **5. 轮播图**





### **5.1 containers/Home/index.tsx**

src/containers/Home/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** Header **from** './components/Header';**import** {Store} **from** '../../types';**import** actions **from** '../../store/actions/home';**import** Swiper **from** './components/Swiper';**import** './index.less';

interface Props{

category:string,

changeCategory:any,

sliders:string[],

getSliders:any

}**class** **Home** **extends** **React**.**Component**<**Props**>{

mainContent:any

componentDidMount(){

**this**.props.getSliders();

}

render(){

**return** (

<React.Fragment>

<Header

category={this.props.category}

changeCategory={this.props.changeCategory}

/>

<div className="main-content" ref={ref=>this.mainContent=ref}>

<Swiper

sliders={this.props.sliders}

/>

</div>

</React.Fragment>

)

}

}**export** **default** connect(

(state:Store)=>state.home,

actions

)(Home);

### **5.2 containers\Home\index.less**

src\containers\Home\index.less

.main-content{

position:fixed;

top:56px;

bottom:54px;

width:100%;

overflow-y: scroll;

overflow-x: hidden;

}

### **5.3 store/action-types.tsx**

src/store/action-types.tsx

//保存当前的轮播图数据**export** **const** SET\_HOME\_SLIDERS = 'SET\_HOME\_SLIDERS';

### **5.4 src/store/actions/home.tsx**

src/store/actions/home.tsx

**import** \* **as** types **from** '../action-types';**import** {getSliders} **from** '../../api/home';**export** interface changeCategory {

type:string,//改变当前的分类

payload:any //新的分类的名称

}//type是用来给类型起别名的**export** type Action = changeCategory;**export** **default** {

changeCategory(category:string):changeCategory{

**return** {type:types.CHANGE\_CATEGORY,payload:category};

},

getSliders(){

**return** **function**(dispatch:any,getState:any){

getSliders().then((sliders:string[])=>{

dispatch({

type:types.SET\_HOME\_SLIDERS,

payload:sliders

});

});

}

},

}

### **5.5 store/reducers/home.tsx**

src/store/reducers/home.tsx

**import** {Home} **from** '../../types';**import** {Action} **from** '../actions/home';**import** \* **as** types **from** '../action-types';**let** initState:Home = {

category:'all',

sliders:[]

};**export** **default** **function**(state:Home=initState,action:Action){

**switch**(action.type){

**case** types.CHANGE\_CATEGORY:

**return** {...state,category:action.payload};

**case** types.SET\_HOME\_SLIDERS:

**return** {...state,sliders:action.payload};

**default**:

**return** state;

}

}

### **5.6 src/types/index.tsx**

src/types/index.tsx

**export** interface Store{

home:Home,

router:any

}**export** interface Home{

category:string,

sliders:string[]

}

**export** interface Props{

children?: any,

}

### **5.7 Home\components\Swiper\index.tsx**

src\containers\Home\components\Swiper\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';**import** \* **as** ReactSwipe **from** 'react-swipe';

interface IProps{

sliders:any

}

interface IState{

index:number

}**class** **Swiper** **extends** **React**.**Component**<**IProps**,**IState**>{

state = {index:0}

render(){

**let** swipeOptions = {

auto: 1000,

continuous: true

}

**let** swipers = (

<ReactSwipe className="carousel" swipeOptions={swipeOptions}>

{

this.props.sliders.map((item:string,index:number)=>(

<div key={index}>

<img src={item}/>

</div>

))

}

</ReactSwipe>

)

**return** (

<div className="home-swipers">

{this.props.sliders.length>0&&swipers}

</div>

)

}

}**export** **default** connect()(Swiper);

### **5.8 containers\Home\components\Swiper\index.less**

src\containers\Home\components\Swiper\index.less

.home-swipers{

position: relative;

img{

width:100%;

}

.dots{

width:100%;

position: absolute;

bottom:10px;

display:flex;

flex-direction: row;

justify-content: center;

align-items: center;

.dot{

width:8px;

height:8px;

border-radius: 50%;

background-color: #FFF;

margin-left:5px;

&.active{

background-color: gray;

}

}

}

}

### **5.9 api\index.tsx**

src\api\index.tsx

**const** API\_HOST='http://localhost:3000';**export** **const** **get**=(url:string) => {

**return** fetch(API\_HOST+url,{

method: 'GET',

credentials: 'include',//跨域携带cookie

headers: {

accept:'application/json'

}

}).then(res=>res.json());

}**export** **const** post=(url:string,data:object) => {

**return** fetch(API\_HOST+url,{

method: 'POST',

body: JSON.stringify(data),

headers: {

'Content-Type': 'application/json',

'Accept':'application/json'

}

});

}

### **5.10 api\home.tsx**

src\api\home.tsx

**import** {**get**} from './index';

export const getSliders=() => {

**return** **get**('/sliders');

}

### **5.11 server\app.js**

server\app.js

**let** express=require('express');**let** app=express();

app.use(**function** (req,res,next) {

res.header('Access-Control-Allow-Methods','PUT,POST,GET,DELETE,OPTIONS');

res.header('Access-Control-Allow-Origin','http://localhost:8080');

res.header('Access-Control-Allow-Credentials','true');

**if** (req.method === 'OPTIONS') {

**return** res.sendStatus(200);

}

next();

});

app.listen(3000);**let** sliders=require('./mock/sliders');

app.get('/sliders',**function** (req,res) {

res.json(sliders);

});

### **5.12 server\mock\sliders.js**

server\mock\sliders.js

module.exports = [

'http://www.zhufengpeixun.cn/themes/jianmo2/images/reactnative.png',

'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

'http://www.zhufengpeixun.cn/themes/jianmo2/images/wechat.png',

'http://www.zhufengpeixun.cn/themes/jianmo2/images/architect.jpg'

];

## **6. 课程列表**

### **6.1 server/mock/lessons.js**

server/mock/lessons.js

module.exports = [

{

id: 1,

title: '1.React全栈架构',

"video":"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥100.00元',

category:'react'

},

{

id: 2,

title: '2.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥200.00元',

category:'react'

},

{

id: 3,

title: '3.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥300.00元',

category:'react'

},

{

id: 4,

title: '4.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥400.00元',

category:'react'

},

{

id: 5,

title: '5.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥500.00元',

category:'react'

},

{

id: 6,

title: '6.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥100.00元',

category:'vue'

},

{

id: 7,

title: '7.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥200.00元',

category:'vue'

},

{

id: 8,

title: '8.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥300.00元',

category:'vue'

},

{

id: 9,

title: '9.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥400.00元',

category:'vue'

},

{

id: 10,

title: '10.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥500.00元',

category:'vue'

},

{

id: 11,

title: '11.React全栈架构',

"video":"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥600.00元',

category:'react'

},

{

id: 12,

title: '12.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥700.00元',

category:'react'

},

{

id: 13,

title: '13.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥800.00元',

category:'react'

},

{

id: 14,

title: '14.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥900.00元',

category:'react'

},

{

id: 15,

title: '15.React全栈架构',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/react/img/react.jpg",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/react.png',

price: '¥1000.00元',

category:'react'

},

{

id: 16,

title: '16.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥600.00元',

category:'vue'

},

{

id: 17,

title: '17.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥700.00元',

category:'vue'

},

{

id: 18,

title: '18.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥800.00元',

category:'vue'

},

{

id: 19,

title: '19.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥900.00元',

category:'vue'

},

{

id: 20,

title: '20.Vue从入门到项目实战',

video:"http://7xil5b.com1.z0.glb.clouddn.com/zhufengpeixun.mp4",

poster:"http://www.zhufengpeixun.cn/vue/img/vue.png",

url: 'http://www.zhufengpeixun.cn/themes/jianmo2/images/vue.png',

price: '¥1000.00元',

category:'vue'

}

]

### **6.2 server/app.js**

server/app.js

**let** express = require('express');**let** bodyParser = require('body-parser');**let** session = require('express-session');//会话可 保存在很多地方 比内存，数据库redis mysql mongodb 文件//let RedisStore = require('connect-redis')(session);**let** app = express();

app.use(bodyParser.urlencoded({extended:true}));//解析form格式的请求体

app.use(bodyParser.json());//解析 json格式的请求

app.use(session({

secret:'zfpx',

resave:true,

cookie:{

maxAge:60\*60\*1000

},

// store: new RedisStore({url:'http://localhost:6379'}),

saveUninitialized:true

}));

app.listen(3000);

app.use(**function**(req,res,next){

res.header('Access-Control-Allow-Origin','http://localhost:8080');

res.header('Access-Control-Allow-Methods','GET,POST,OPTIOINS');

res.header('Access-Control-Allow-Headers','Accept,Content-Type');

res.header('Access-Control-Allow-Credentials','true');//允许客户端跨域发cookie

**if**(req.method == 'options'){

res.end('');

}**else**{

next();

}

});**let** sliders = require('./mock/sliders');

app.get('/api/sliders',**function**(req,res){

res.json(sliders);

});**let** lessons = require('./mock/lessons');// '/api/lessons/react?offset=0&limit=5

app.get('/api/lessons/:category',**function**(req,res){

**let** data = JSON.parse(JSON.stringify(lessons));

**let** category = req.params.category;//取得分类名称

**let** offset = req.query.offset;

**let** limit = req.query.limit;

offset = isNaN(offset)?0:parseInt(offset);

limit = isNaN(limit)?5:parseInt(limit);

//先拿条件过滤一下

**if**(category != 'all'){

data = data.filter(item=>item.category == category);

}

//pageSize pageNumber

// 10 第一页 0 5 第二页 5 10

**let** list = data.slice(offset,offset+limit);//包前不包后 本页的条数

**let** hasMore = data.length > offset+limit; //是否还有更多

setTimeout(**function**(){

res.json({code:0,data:{list,hasMore}});

/\* if(Math.random()>.5){// code是1还是0来判断请求是成功还是失败

res.json({code:0,data:{list,hasMore}});

}else{

res.json({code:1,error:'数据加载失败'});

} \*/

},1000);

});

### **6.3 api/home.tsx**

src/api/home.tsx

**import** {**get**} from './index';

export const getSliders = ()=>{

**return** **get**('/api/sliders');

}

export const getLessons = (category:string,offset:number,limit:number)=>{

**return** **get**(`/api/lessons/${category}?offset=${offset}&limit=${limit}`);

}

### **6.4 Home/components/Header/index.tsx**

src/containers/Home/components/Header/index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** { CSSTransition,TransitionGroup } **from** 'react-transition-group';

declare **function** **require**(url: string): **string**;**const** **logo**=**require**('../../../../images/logo.png');**interface** **Props**{

category:string,

changeCategory:any,

refreshLessons:any

}

interface IState{

showList:any

}**export** **default** **class** **HomeHeader** **extends** **React**.**Component**<**Props**,**IState**> {

state={

showList:false,

}

changeCategory = (event:any)=>{

**let** category = event.target.dataset.category;

**this**.setState({showList:false},()=>{

**this**.props.changeCategory(category);

**this**.props.refreshLessons();

});

}

render() {

**let** {category} = **this**.props;

**return** (

<div className="home-header">

<div className="header-menu">

<img src={logo} alt="logo" />

<div onClick={() => this.setState({showList:!this.state.showList})}>

{

this.state.showList?<i className="iconfont icon-guanbi"></i>:<i className="iconfont icon-uilist"></i>

}

</div>

</div>

<TransitionGroup>

{

this.state.showList&&<CSSTransition

timeout={500}

classNames="fade"

>

<ul className="menu-list" onClick={this.changeCategory}>

<li data-category="react" className={category=='react'?'active':''}>React</li>

<li data-category="vue" className={category=='vue'?'active':''}>Vue</li>

</ul>

</CSSTransition>

}

</TransitionGroup>

</div>

);

}

}

### **6.5 containers/Home/index.tsx**

src/containers/Home/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** Header **from** './components/Header';**import** {Store} **from** '../../types';**import** actions **from** '../../store/actions/home';**import** Swiper **from** './components/Swiper';**import** List **from** './components/List';**import** './index.less';

interface Props{

category:string,

changeCategory:any,

sliders:string[],

getSliders:any,

lessons:any,

getLessons:any,

refreshLessons:any

}**class** **Home** **extends** **React**.**Component**<**Props**>{

mainContent:any

componentDidMount(){

**this**.props.getSliders();

**this**.props.getLessons();

}

render(){

**return** (

<React.Fragment>

<Header

category={this.props.category}

changeCategory={this.props.changeCategory}

refreshLessons={this.props.refreshLessons}

/>

<div className="main-content" ref={ref=>this.mainContent=ref}>

<Swiper

sliders={this.props.sliders}

/>

<List

lessons={this.props.lessons}

getLessons={this.props.getLessons}

/>

</div>

</React.Fragment>

)

}

}**export** **default** connect(

(state:Store)=>state.home,

actions

)(Home);

### **6.6 store/action-types.tsx**

src/store/action-types.tsx

//是用来把lessons里的loading状态改为true**export** **const** GET\_HOME\_LESSONS\_LOADING = 'GET\_HOME\_LESSONS\_LOADING';//当响应回来后，把服务 器的数据放置到仓库中，顺便 把 loading设置为false**export** **const** SET\_HOME\_LESSONS = 'SET\_HOME\_LESSONS';//是用来把lessons里的loading状态改为true**export** **const** REFRESH\_HOME\_LESSONS\_LOADING = 'REFRESH\_HOME\_LESSONS\_LOADING';**export** **const** REFRESH\_HOME\_LESSONS = 'REFRESH\_HOME\_LESSONS';

### **6.7 store/actions/home.tsx**

src/store/actions/home.tsx

**import** \* **as** types **from** '../action-types';**import** {getSliders,getLessons} **from** '../../api/home';**export** interface changeCategory {

type:string,//改变当前的分类

payload:any //新的分类的名称

}//type是用来给类型起别名的**export** type Action = changeCategory;**export** **default** {

changeCategory(category:string):changeCategory{

**return** {type:types.CHANGE\_CATEGORY,payload:category};

},

getSliders(){

**return** **function**(dispatch:any,getState:any){

getSliders().then((sliders:string[])=>{

dispatch({

type:types.SET\_HOME\_SLIDERS,

payload:sliders

});

});

}

},

getLessons(){

**return** **function**(dispatch:any,getState:any){

**let** {category,lessons:{hasMore,loading,offset,limit}} = getState().home;

**if**(hasMore && !loading){//如果有下一页数据并且不当不是处于加载中的话才会发请求

dispatch({type:types.GET\_HOME\_LESSONS\_LOADING,payload:true});//loading=true

getLessons(category,offset,limit).then(result =>{

**let** {code,data,error} = result;

**if**(code ==0){

dispatch({type:types.SET\_HOME\_LESSONS,payload:data});

}**else**{

dispatch({type:types.GET\_HOME\_LESSONS\_LOADING,payload:false});//loading=true

alert(error);

}

});

}

}

},

//重新查询

refreshLessons(){

**return** **function**(dispatch:any,getState:any){

**let** {category,lessons:{hasMore,loading,offset,limit}} = getState().home;

**if**(!loading){//如果有下一页数据并且不当不是处于加载中的话才会发请求

dispatch({type:types.REFRESH\_HOME\_LESSONS\_LOADING,payload:true});//loading=true

getLessons(category,0,limit).then(result =>{

**let** {code,data,error} = result;

**if**(code ==0){

dispatch({type:types.REFRESH\_HOME\_LESSONS,payload:data});

}**else**{

dispatch({type:types.REFRESH\_HOME\_LESSONS\_LOADING,payload:false});//loading=true

alert(error);

}

});

}

}

}

}

### **6.8 store/reducers/home.tsx**

src/store/reducers/home.tsx

**import** {Home} **from** '../../types';**import** {Action} **from** '../actions/home';**import** \* **as** types **from** '../action-types';**let** initState:Home = {

category:'all',

sliders:[],

lessons:{

list:[],

hasMore:true,

loading:false,

offset:0,

limit:5

}

};**export** **default** **function**(state:Home=initState,action:Action){

**switch**(action.type){

**case** types.CHANGE\_CATEGORY:

**return** {...state,category:action.payload};

**case** types.SET\_HOME\_SLIDERS:

**return** {...state,sliders:action.payload};

**case** types.GET\_HOME\_LESSONS\_LOADING://只修改loading状态

**return** {...state,lessons:{...state.lessons,loading:action.payload}};

**case** types.SET\_HOME\_LESSONS:

**return** {...state,lessons:{

...state.lessons,

list:[...state.lessons.list,...action.payload.list],

hasMore:action.payload.hasMore,

loading:false,

offset:state.lessons.offset+action.payload.list.length

}};

**case** types.REFRESH\_HOME\_LESSONS\_LOADING:

**return** {...state,lessons:{

...state.lessons,

list:[],

offset:0,

hasMore:true,

loading:action.payload

}};

**case** types.REFRESH\_HOME\_LESSONS:

**return** {...state,lessons:{

...state.lessons,

list:action.payload.list,

hasMore:action.payload.hasMore,

loading:false,

offset:action.payload.list.length

}};

**default**:

**return** state;

}

}

### **6.9 types/index.tsx**

src/types/index.tsx

**export** interface Store{

home:Home,

router:any

}**export** interface Home{

category:string,

sliders:string[],

lessons:Lessons

}

**export** interface Props{

children?: any,

}

**export** interface Lessons{

list:any[],//每页的数据

hasMore:boolean,//是否有更多

offset:number,//偏移量

limit:number,//每页的条数

loading:boolean//当前是否正在加载

}

### **6.10 components\Loading\index.tsx**

src\components\Loading\index.tsx

import \* as React from 'react';

import './index.less';

declare function require(url:string):string;

let loading = require('../../images/loading.gif');

export default ()=> (

<div className="loading">

<img src={loading}/>

</div>

)

### **6.11 components\Loading\index.less**

src\components\Loading\index.less

.loading{

width:100%;

height:37px;

line-height: 37px;

text-align: center;

img{

width:37px;

height:37px;

}

}

### **6.12 Home\components\List\index.tsx**

src\containers\Home\components\List\index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** Loading **from** '../../../../components/Loading';**import** {Link} **from** 'react-router-dom';**import** './index.less';

interface Props{

lessons:any,

getLessons:any

}**class** **List** **extends** **React**.**Component**<**Props**>{

render(){

**let** {list,hasMore,loading} = **this**.props.lessons;

**return** (

<div className="home-lessons">

<div className="all-lessons">

<i className="iconfont icon-kecheng-copy"></i>

<span>全部课程</span>

</div>

{

list.length>0?list.map((item:any,index:number):any=>(

<Link key={index} to={{pathname:`/detail/${item.id}`,state:item}}>

<div className="lesson" >

<img src={item.poster} alt={item.title}/>

<p>{item.title}</p>

<p>{item.price}</p>

</div>

</Link>

)):<div className="nodata">暂无数据</div>

}

{

loading?<Loading/>:<div className="load-more" onClick={this.props.getLessons}>加载更多</div>

}

</div>

)

}

}**export** **default** connect()(List);

### **6.13 Home\components\List\index.less**

src\containers\Home\components\List\index.less

.home-lessons{

width:100%;

box-sizing: border-box;

padding: 7.5px;

.all-lessons{

margin:10px 0;

span{

margin-left:8px;

}

}

.nodata{

width:100%;

height:35px;

line-height: 35px;

border-radius: 5px;

color:#000;

text-align: center;

}

.lesson{

width:100%;

border-radius: 8px;

margin-bottom: 17px;

box-shadow: 1px 1px 3px 2px #c5c5c5,-1px -1px 3px 2px #c5c5c5;

overflow: hidden;

img{

width:100%;

height:140px;

border-radius: 8px 8px 0 0;

}

p{

height:37px;

text-align: center;

line-height: 37px;

&:nth-child(2){

color:#777777;

}

&:nth-child(3){

color:#ed3a3a;

}

}

}

.load-more{

width:100%;

height:35px;

line-height: 35px;

border-radius: 5px;

color:#FFF;

background-color: green;

text-align: center;

}

}

## **7. 上拉加载和记录滚动条位置**

### **7.1 src/utils.tsx**

**export** **function** **loadMore**(element:any,callback:any){

//防抖 节流

**let** timer:any;

element.addEventListener('scroll',**function**(){

timer&&clearTimeout(timer);

timer = setTimeout(**function**(){

**let** clientHeight = element.clientHeight;//div的高度，可视区域 的高度

**let** scrollTop = element.scrollTop;//向上卷去的高度

**let** scrollHeight = element.scrollHeight;//内容 高度

**if**(clientHeight + scrollTop+10 >= scrollHeight){

callback();

}

},300);

});

}

**export** **function** **downRefresh**(element:any,callback:any){

**let** startY:number;//按下时候的初始纵坐标

**let** distance:number;//一共下拉的距离

**let** originalTop = element.offsetTop;//最初的元素距离 父级顶部的距离

element.addEventListener('touchstart',**function**(event:any){

**if**(element.offsetTop == originalTop &&element.scrollTop == 0){

startY = event.touches[0].pageY;

element.addEventListener('touchmove',touchMove);

element.addEventListener('touchend',touchEnd);

}

});

**function** **touchMove**(event:any){

**let** pageY = event.touches[0].pageY;

**if**(pageY>startY){

distance = pageY - startY;

element.style.top = (originalTop+distance)+'px';

}**else**{

element.removeEventListener('touchmove',touchMove);

element.removeEventListener('touchend',touchEnd);

}

}

**function** **touchEnd**(event:any){

element.removeEventListener('touchmove',touchMove);

element.removeEventListener('touchend',touchEnd);

**let** timer = setInterval(**function**(){

**if**(distance < 1){

element.style.top = originalTop+'px';

clearInterval(timer);

}**else**{

element.style.top = (originalTop+--distance)+'px';

}

},13);

**if**(distance>10){

callback();

}

}

}//封装了一个工具方法，用来往sessionStorage存值和取值 **export** **const** store = {

**set**(key:string,val:string){

sessionStorage.setItem(key,val);

},

**get**(key:string){

**return** sessionStorage.getItem(key);

}

}

### **7.2 Home/index.tsx**

src/containers/Home/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** Header **from** './components/Header';**import** {Store} **from** '../../types';**import** actions **from** '../../store/actions/home';**import** Swiper **from** './components/Swiper';**import** List **from** './components/List';**import** {loadMore,downRefresh,store} **from** '../../utils';**import** './index.less';

interface Props{

category:string,

changeCategory:any,

sliders:string[],

getSliders:any,

lessons:any,

getLessons:any,

refreshLessons:any

}**class** **Home** **extends** **React**.**Component**<**Props**>{

mainContent:any

componentDidMount(){

**if**(**this**.props.sliders.length >0){

**this**.mainContent.scrollTop = store.get('homeScrollTop');

}**else**{

**this**.props.getSliders();

**this**.props.getLessons();

}

loadMore(**this**.mainContent,**this**.props.getLessons);

downRefresh(**this**.mainContent,**this**.props.refreshLessons);

}

//在组件将要被销毁的时候

componentWillUnmount(){

store.set('homeScrollTop',**this**.mainContent.scrollTop);

}

render(){

**return** (

<React.Fragment>

<Header

category={this.props.category}

changeCategory={this.props.changeCategory}

refreshLessons={this.props.refreshLessons}

/>

<div className="main-content" ref={ref=>this.mainContent=ref}>

<Swiper

sliders={this.props.sliders}

/>

<List

lessons={this.props.lessons}

getLessons={this.props.getLessons}

/>

</div>

</React.Fragment>

)

}

}**export** **default** connect(

(state:Store)=>state.home,

actions

)(Home);

### **7.3 Home/components/List/index.tsx**

src/containers/Home/components/List/index.tsx

{

loading?<Loading/>:( !hasMore&&<div className="load-more">我是有底线的</div>)

}

## **8.课程详情页**

### **8.1 src/containers/App.tsx**

src/containers/App.tsx

**import** {Route,Link,Switch} **from** 'react-router-dom';**import** \* **as** React **from** 'react';**import** Tab **from** '../components/Tab';**import** Home **from** './Home';**import** Mime **from** './Mime';**import** Profile **from** './Profile';**import** Detail **from** './Detail';**import** '../common/index.less';

interface IProps{

children:any

}**export** **default** **class** **App** **extends** **React**.**Component**<**IProps**>{

render(){

**return** (

<React.Fragment>

<Route exact path="/" component={Home}/>

<Route path="/mime" component={Mime}/>

<Route path="/profile" component={Profile}/>

<Route path="/detail/:id" component={Detail} />

<Tab></Tab>

</React.Fragment>

)

}

}

### **8.2 NavHeader\index.tsx**

src\components\NavHeader\index.tsx

**import** \* **as** React **from** 'react';**import** './index.less';

interface Props{

history: any,

title: string

}**export** **default** **class** **NavHeader** **extends** **React**.**Component**<**Props**>{

render(){

**return** (

<div className="navheader">

<i

onClick={()=>this.props.history.goBack()}

className="iconfont icon-fanhui"></i>

{this.props.title}

</div>

)

}

}

### **8.3 NavHeader\index.less**

src\components\NavHeader\index.less

.navheader{

height:56px;

background-color: #000;

color:#FFF;

line-height: 56px;

text-align: center;

position: fixed;

top:0;

left:0;

width:100%;

i{

position: absolute;

left:10px;

}

}

### **8.4 containers\Detail\index.tsx**

src\containers\Detail\index.tsx

**import** \* **as** React **from** 'react';**import** NavHeader **from** '../../components/NavHeader';**import** {Redirect} **from** 'react-router-dom';**import** './index.less';

interface Props{

location: any,

history:any,

match:any

}**export** **default** **class** **Detail** **extends** **React**.**Component**<**Props**>{

state = {

lesson:**this**.props.location.state||{}

}

render(){

**let** { lesson }=**this**.state;

**return** (

lesson?(

<div className="lesson-detail">

<NavHeader title="课程详情" history={this.props.history} />

<img src={lesson.poster}/>

<p>{lesson.title}</p>

<p>{lesson.price}</p>

</div>

):<Redirect to="/"/>

)

}

}

### **8.5 containers\Detail\index.less**

src\containers\Detail\index.less

.lesson-detail{

padding-top:56px;

img{

height:167px;

width:100%;

}

p{

text-align: center;

line-height:34px;

height:34px;

&:nth-child(2){

color:#CCC;

}

&:nth-child(2){

color:#F00;

}

}

}

## **9.个人中心**

### **9.1 Profile\index.tsx**

src\containers\Profile\index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** { Link } **from** 'react-router-dom';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**export** **default** **class** **Profile** **extends** **React**.**Component**{

render() {

**return** (

<div className="profile">

<div className="profile-bg">

<img src={profile}/>

<div className="login-btn">

<Link to="/login">登录</Link>

</div>

</div>

</div>

)

}

}

### **9.2 containers\Profile**

src\containers\Profile\index.less

.profile{

.profile-bg{

width:100%;

height:223px;

background-image: url(../../images/login\_bg.png);

background-size:contain;

display:flex;

flex-direction: column;

justify-content: center;

align-items: center;

img{

width:60px;

height:60px;

border-radius:50%;

}

.login-btn{

width:60px;

height:25px;

border-radius: 10px;

background:#FFF;

color:#188AE4;

line-height:25px;

text-align:center;

font-size:13px;

font-weight: bold;

margin-top:10px;

a{

text-decoration: none;

&:link{

text-decoration: none;

color:#188AE4;

}

}

}

}

}

## **10.登录**

### **10.1 src/containers/App.tsx**

src/containers/App.tsx

+ import Login from './Login';+ <Route path="/login" component={Login}/>

### **10.2 containers\Login\index.tsx**

src\containers\Login\index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** {Link} **from** 'react-router-dom';**import** NavHeader **from** '../../components/NavHeader';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

history:any

}**export** **default** **class** **Login** **extends** **React**.**Component**<**Props**>{

render() {

**return** (

<div className="login-panel">

<NavHeader title="登录" history={this.props.history} />

<div className="login-logo">

<img src={profile} />

</div>

<input type="text" placeholder="手机号" />

<input type="text" placeholder="密码" />

<Link to="/reg">前往注册</Link>

<button>登&nbsp;录</button>

</div>

)

}

}

### **10.3 containers\Login\index.less**

src\containers\Login\index.less

.login-panel{

padding:56px 17px;

display:flex;

flex-direction: column;

position: absolute;

top:0;

left:0;

right:0;

bottom:0;

background:#FFF;

z-index: 20;

.login-logo{

display:flex;

height:223px;

width:100%;

justify-content: center;

align-items: center;

img{

height:62px;

width:62px;

}

}

input{

width:100%;

outline:none;

border:1px solid #CCC;

border-radius: 5px;

height:42px;

margin-bottom:20px;

padding-left:10px;

box-sizing: border-box;

}

a{

color:#188AE4;

margin-bottom:20px;

}

button{

width:100%;

border:none;

outline:none;

background-color: #188AE4;

color:#FFF;

height:40px;

line-height: 40px;

text-align:center;

font-size:16px;

border-radius:5px;

}

}

## **11.注册**

### **11.1 src/containers/App.tsx**

src/containers/App.tsx

+ import Reg from './Reg';+ <Route path="/reg" component={Reg}/>

### **11.2 Reg\index.tsx**

src\containers\Reg\index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** {Link} **from** 'react-router-dom';**import** NavHeader **from** '../../components/NavHeader';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

history:any

}**export** **default** **class** **Reg** **extends** **React**.**Component**<**Props**>{

render() {

**return** (

<div className="login-panel">

<NavHeader title="注册" history={this.props.history}/>

<div className="login-logo">

<img src={profile} />

</div>

<input type="text" placeholder="手机号" />

<input type="text" placeholder="密码" />

<Link to="/login">前往登录</Link>

<button>注&nbsp;册</button>

</div>

)

}

}

### **11.3 Reg\index.less**

src\containers\Reg\index.less

.login-panel{

padding:56px 17px;

display:flex;

flex-direction: column;

position: absolute;

top:0;

left:0;

right:0;

bottom:0;

background:#FFF;

z-index: 20;

.login-logo{

display:flex;

height:223px;

width:100%;

justify-content: center;

align-items: center;

img{

height:62px;

width:62px;

}

}

input{

width:100%;

outline:none;

border:1px solid #CCC;

border-radius: 5px;

height:42px;

margin-bottom:20px;

padding-left:10px;

box-sizing: border-box;

}

a{

color:#188AE4;

margin-bottom:20px;

}

button{

width:100%;

background-color: #188AE4;

color:#FFF;

height:40px;

line-height: 40px;

text-align:center;

font-size:16px;

border-radius:5px;

}

}

## **12.实现会话**

### **12.1 server/app.js**

server/app.js

**let** express=require('express');**let** bodyParser = require('body-parser');**let** session=require('express-session');**let** app=express();

app.use(bodyParser.json());

app.use(**function** (req,res,next) {

res.header('Access-Control-Allow-Methods','PUT,POST,GET,DELETE,OPTIONS');

res.header('Access-Control-Allow-Headers','Content-Type');

res.header('Access-Control-Allow-Origin','http://localhost:8080');

res.header('Access-Control-Allow-Credentials','true');

**if** (req.method === 'OPTIONS') {

**return** res.sendStatus(200);

}

next();

});

app.use(session({

resave:true,

secret:'zfpx',

saveUninitialized:true

}));

app.listen(3000);**let** sliders=require('./mock/sliders');

app.get('/api/sliders',**function** (req,res) {

res.json(sliders);

});**let** lessons = require('./mock/lessons');// http://getLessons/vue?offset=0&limit=5

app.get('/api/lessons/:category',**function**(req,res){

**let** category = req.params.category;

**let** {offset,limit} = req.query;

offset = isNaN(offset)?0:parseInt(offset);//偏移量

limit = isNaN(limit)?5:parseInt(limit); //每页条数

**let** list = JSON.parse(JSON.stringify(lessons));

**if**(category!='all'){

list = list.filter(item=>item.category==category);

}

**let** total = list.length;

//分页数据

list = list.slice(offset,offset+limit);

//list.forEach(item=>item.title= item.title+Math.random());

setTimeout(**function**(){

res.json({

code:0,

data:{

list,

hasMore:total>offset+limit

}

});

},1000);

});

// 30 6次 5条 30 offset=30 **let** users = [];

app.post('/api/reg',**function**(req,res){

**let** body = req.body;

users.push(body);

res.json({

code:0,

success:'注册成功'

})

});//user username password

app.post('/api/login',**function**(req,res){

**let** body = req.body;//{username,password}

**let** user = users.find(item=>item.username == body.username && item.password == body.password);

**if**(user){

req.session.user = user;

res.json({

user,

code:0,

success:'登录成功'

});

}**else**{

res.json({

code:1,

error:'登录失败'

});

}

});

app.get('/api/validate',**function**(req,res){

**let** user = req.session.user;

**if**(user){

res.json({

code:0,

success:'此用户已经登录',

user

});

}**else**{

res.json({

code:1,

error:'此用户未登录',

});

}

});

### **12.2 src/api/index.tsx**

src/api/index.tsx

**const** API\_HOST='http://localhost:3000';**export** **const** **get**=(url:string) => {

**return** fetch(API\_HOST+url,{

method: 'GET',

credentials: 'include',//跨域携带cookie

headers: {

accept:'application/json'

}

}).then(res=>res.json());

}**export** **const** post=(url:string,data:object) => {

**return** fetch(API\_HOST+url,{

method: 'POST',

body: JSON.stringify(data),

headers: {

'Content-Type': 'application/json',

'Accept':'application/json'

}

}).then(res=>res.json());

}

### **12.3 Login/index.tsx**

src/containers/Login/index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** {Link} **from** 'react-router-dom';**import** NavHeader **from** '../../components/NavHeader';**import** { connect } **from** 'react-redux';**import** { Store } **from** '../../types';**import** actions **from** '../../store/actions/session';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

history: any,

login:any

}**class** **Login** **extends** **React**.**Component**<**Props**>{

username:any

password:any

handleLogin = ()=>{

**let** username = **this**.username.value;

**let** password = **this**.password.value;

**this**.props.login({username,password});

}

render() {

**return** (

<div className="login-panel">

<NavHeader title="登录" history={this.props.history} />

<div className="login-logo">

<img src={profile} />

</div>

<input ref={input=>this.username=input} type="text" placeholder="手机号" />

<input ref={input=>this.password=input} type="text" placeholder="密码" />

<Link to="/reg">前往注册</Link>

<button onClick={this.handleLogin}>登&nbsp;录</button>

</div>

)

}

}

**export** **default** connect(

(state:Store)=>state.session,actions

)(Login);

### **12.4 Profile/index.tsx**

src/containers/Profile/index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** { Link } **from** 'react-router-dom';**import** { Store } **from** '../../types';**import** { connect } **from** 'react-redux';**import** actions **from** '../../store/actions/session';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

user:any

}**class** **Profile** **extends** **React**.**Component**<**Props**>{

render() {

**return** (

<div className="profile">

<div className="profile-bg">

<img src={profile}/>

<div className="login-btn">

{this.props.user?this.props.user.username:<Link to="/login">登录</Link>}

</div>

</div>

</div>

)

}

}**export** **default** connect(

(state:Store)=>state.session,actions

)(Profile);

### **12.5 containers/Reg/index.tsx**

src/containers/Reg/index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** {Link} **from** 'react-router-dom';**import** NavHeader **from** '../../components/NavHeader';**import** { connect } **from** 'react-redux';**import** { Store } **from** '../../types';**import** actions **from** '../../store/actions/session';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

history: any,

reg:any

}**class** **Login** **extends** **React**.**Component**<**Props**>{

username: any

password: any

handleReg = ()=>{

**let** username = **this**.username.value;

**let** password = **this**.password.value;

**this**.props.reg({username,password});

}

render() {

**return** (

<div className="login-panel">

<NavHeader title="注册" history={this.props.history}/>

<div className="login-logo">

<img src={profile} />

</div>

<input ref={input=>this.username=input} type="text" placeholder="手机号" />

<input ref={input=>this.password=input} type="text" placeholder="密码" />

<Link to="/login">前往登录</Link>

<button onClick={this.handleReg} >注&nbsp;册</button>

</div>

)

}

}

**export** **default** connect(

(state:Store)=>state.session,actions

)(Login);

### **12.6 store/action-types.tsx**

src/store/action-types.tsx

//注册**export** **const** REG = 'REG';//登录**export** **const** LOGIN = 'LOGIN';//退出**export** **const** LOGOUT = 'LOGOUT';

//清空消息**export** **const** CLEAR\_MESSAGES = 'CLEAR\_MESSAGES';**export** **const** VALIDATE = 'VALIDATE';

### **12.7 store/reducers/index.tsx**

src/store/reducers/index.tsx

**import** { combineReducers } **from** 'redux';**import** history **from** '../history';**import** home **from** './home';**import** session **from** './session';**import** { connectRouter } **from** 'connected-react-router'**let** reducers=combineReducers({

router: connectRouter(history),

home,

session,

});**export** **default** reducers;

### **12.8 types/index.tsx**

src/types/index.tsx

**export** interface Store{

home:Home,

session:Session,

router:any

}**export** interface Session{

user?: any,

error: any,

success:any

}**export** interface Home{

category:string,

sliders:string[],

lessons:Lessons

}

**export** interface Props{

children?: any,

}

**export** interface Lessons{

list:any[],//每页的数据

hasMore:boolean,//是否有更多

offset:number,//偏移量

limit:number,//每页的条数

loading:boolean//当前是否正在加载

}

### **12.9 src/api/session.tsx**

src/api/session.tsx

**import** {**get**,post} from './index';

//注册

export function reg(user:any){

**return** post('/api/reg',user);//{username,password}

}//登录**export** **function** **login**(user:any){

**return** post('/api/login',user);//{username,password}

}//退出**export** **function** **logout**(){

**return** **get**('/api/logout');

}

export function validate(){

**return** **get**('/api/validate');

}

### **12.10 store/actions/session.tsx**

src/store/actions/session.tsx

**import** \* **as** types **from** '../action-types';**import** { reg,login,logout,validate } **from** '../../api/session';**import** {push} **from** 'connected-react-router';**export** type Action=any;

interface Res{

code:any,success:any,error:any

}**export** **default** {

reg(user:any) {

**return** **function** (dispatch:any,getState:any) {

reg(user).then((result) => {

**let** { code,success,error }=result;

dispatch({

type: types.REG,

payload: { success,error }

});

**if** (code==0) {//code=0表示成功 成功后跳到登录页

dispatch(push('/login'));

}

})

}

},

login(user:any) {

**return** **function** (dispatch:any,getState:any) {

login(user).then(result => {

**let** { code,success,error,user }=result;

dispatch({

type: types.LOGIN,

payload: { success,error,user }

});

**if** (code==0) {

dispatch(push('/profile'));

}

})

}

},

logout() {

**return** **function** (dispatch:any,getState:any) {

logout().then(result => {

**let** { code,success,error }=result;

dispatch({

type: types.LOGOUT,

payload: { success,error }

});

dispatch(push('/login'));

});

}

},

clearMessages() {

**return** {

type: types.CLEAR\_MESSAGES

}

},

validate() {

**return** **function** (dispatch:any,getState:any) {

validate().then(result => {

**let** { code,success,error,user }=result;

dispatch({

type: types.VALIDATE,

payload: { success,error,user }

});

});

}

}

}

### **12.11 store/reducers/session.tsx**

src/store/reducers/session.tsx

**import** \* **as** types **from** '../action-types';**import** { Session } **from** '../../types';

**let** initState:Session = {

error: '',//错误消息

success: '',//成功消息

//如果登录成功的话，需要给此属性赋值为登录用户

}**export** **default** **function** (state:Session = initState, action:any) {

**switch** (action.type) {

**case** types.REG:///注册方法调用完成后

//不需要解构老状态

**return** {

...action.payload

};

**case** types.LOGIN:///注册方法调用完成后

**return** {

...action.payload

};

**case** types.LOGOUT:///退出方法调用完成后

**return** {

...action.payload

};

**case** types.CLEAR\_MESSAGES:

**return** {

...state,

error: '',

success:''

};

**case** types.VALIDATE:

**return** {

...state,

...action.payload

};

**default**:

**return** state;

}

}

## **13.受保护的路由**

### **13.1 src/containers/App.tsx**

src/containers/App.tsx

**import** {Route,Link,Switch} **from** 'react-router-dom';**import** \* **as** React **from** 'react';**import** Tab **from** '../components/Tab';**import** Home **from** './Home';**import** Mime **from** './Mime';**import** Profile **from** './Profile';**import** Detail **from** './Detail';**import** Login **from** './Login';**import** Reg **from** './Reg';**import** '../common/index.less';**import** PrivateRoute **from** './PrivateRoute';

interface IProps{

children:any

}**export** **default** **class** **App** **extends** **React**.**Component**<**IProps**>{

render(){

**return** (

<React.Fragment>

<Switch>

<Route exact path="/" component={Home}/>

<Route path="/mime" component={Mime}/>

<Route path="/detail/:id" component={Detail} />

<Route path="/login" component={Login}/>

<Route path="/reg" component={Reg}/>

<PrivateRoute pathname={window.location.pathname} path="/profile" component={Profile} />

</Switch>

<Tab></Tab>

</React.Fragment>

)

}

}

### **13.2 src/containers/Profile/index.tsx**

src/containers/Profile/index.tsx

**import** \* **as** React **from** 'react';**import** './index.less'**import** { Link } **from** 'react-router-dom';**import** { connectRouter } **from** 'connected-react-router';**import** { connect } **from** 'react-redux';**import** { Store } **from** '../../types';

declare **function** **require**(url: string): **string**;**let** **profile**=**require**('../../images/profile.png');**interface** **Props**{

user:any

}

**class** **Profile** **extends** **React**.**Component**<**Props**>{

render() {

**return** (

<div className="profile">

<div className="profile-bg">

<img src={profile}/>

<div className="login-btn">

{this.props.user?this.props.user.username:<Link to="/login">登录</Link>}

</div>

</div>

</div>

)

}

}**export** **default** connect(

(state:Store)=>state.session

)(Profile);

### **13.3 containers\PrivateRoute\index.tsx**

src\containers\PrivateRoute\index.tsx

**import** \* **as** React **from** 'react';**import** {Route,Redirect} **from** 'react-router-dom';**import** { connect } **from** 'react-redux';**import** { Store } **from** '../../types';

interface Props{

path: any,

component: any,

user:any,

pathname?:string

}**class** **PrivateRoute** **extends** **React**.**Component**<**Props**> {

render() {

**let** { path,component: Comp,user }=**this**.props;

**return** <Route path={path} render={(props) => user? <Comp {...props} />:<Redirect to="/login" />} />;

}

}**export** **default** connect(

(state:Store) => state.session

)(PrivateRoute)

## **14.点状导航**

### **14.1 containers/Home/components/Swiper/index.tsx**

src/containers/Home/components/Swiper/index.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';**import** \* **as** ReactSwipe **from** 'react-swipe';**import** SwiperItems **from** './SwiperItems';

interface IProps{

sliders:any

}

interface IState{

index:number

}**class** **Swiper** **extends** **React**.**Component**<**IProps**,**IState**>{

state = {index:0}

changeIndex = (index:number)=>{

**this**.setState({index});

}

render(){

**return** (

<div className="home-swipers">

<SwiperItems sliders={this.props.sliders} changeIndex={this.changeIndex}/>

<div className="dots">

{

this.props.sliders.map((item:string,index:number) => (

<span key={index} className={`dot ${this.state.index ==index?'active':''}`}></span>

))

}

</div>

</div>

)

}

}**export** **default** connect()(Swiper);

### **14.2 Home/components/Swiper/SwiperItems.tsx**

src/containers/Home/components/Swiper/SwiperItems.tsx

**import** \* **as** React **from** 'react';**import** { connect } **from** 'react-redux';**import** './index.less';**import** \* **as** ReactSwipe **from** 'react-swipe';

interface IProps{

sliders:any,

changeIndex:any

}

**class** **Swiper** **extends** **React**.**Component**<**IProps**>{

render(){

**let** swipeOptions = {

auto: 1000,

continuous: true,

callback:(index:number)=>{

**this**.props.changeIndex(index);

}

}

**return** (

<ReactSwipe className="carousel" swipeOptions={swipeOptions}>

{

this.props.sliders.map((item:string,index:number)=>(

<div key={index}>

<img src={item}/>

</div>

))

}

</ReactSwipe>

)

}

}**export** **default** connect()(Swiper);