## **1. 初始化项目 [#](https://zhufeng-document.vercel.app/html/30.cms-14-front.html" \l "t01. %E5%88%9D%E5%A7%8B%E5%8C%96%E9%A1%B9%E7%9B%AE)**

* [create-umi-app](https://umijs.org/zh/guide/create-umi-app.html)
* [ant.design](https://ant.design/docs/react/introduce-cn)

$ mkdir myapp && cd myapp

$ yarn create umi

$ cnpm i styled-components jsonwebtoken -S

$ npm start

## **2. 配置别名**

.webpackrc.js

**const** {resolve}=require('path');**export** **default** {

"alias":{

"@":resolve("src")

}

}

## **3. 注册登录**



### **3.1 login/index.js**

#### **3.1.1 Login**

**import** React,{Component} **from** 'react';**import** {Layout,Form,Input,Radio,Cascader,Select,AutoComplete,Checkbox,Button,message} **from** 'antd';**import** styled **from** 'styled-components';**import** {connect} **from** 'dva';//react-redux 是用来连接仓库和组件**import** addresses **from** '../../utils/addresses';**import** getFieldItems **from** '../../utils/getFieldItems';**const** { Header, Footer, Sider, Content } = Layout;**const** FormItem = Form.Item;**const** {Option} = Select;**class** **Login** **extends** **Component**{

handleSubmit=event => {

event.preventDefault();

**this**.userForm.props.form.validateFields((err,values) => {

**if** (err) {message.error('输入不合法!');

} **else** {

**this**.props.dispatch({type: **this**.props.isLogin?'login/login':'login/signup',payload:values});

}

});

}

changeLoginStatus = ()=>{

**this**.props.dispatch({type:'login/save',payload:{isLogin:!**this**.props.isLogin}});

}

render(){

**return** (

<Layout>

<Content>

<LoginForm

isLogin={this.props.isLogin}

changeLoginStatus={this.changeLoginStatus}

handleSubmit={this.handleSubmit}

wrappedComponentRef={inst => this.userForm=inst}

handleSubmit={this.handleSubmit}

/>

</Content>

</Layout>

)

}

}

#### **3.1.2 LoginForm**

**class** **LoginForm** **extends** **Component**{

state ={gender:1,autoCompleteResult:[],repasswordDirty:false}

handleWebsiteChange = (value)=>{//1 [1.com,1.cn,1.org]

**let** autoCompleteResult=[];

**if**(value){

autoCompleteResult = [".com",".cn",".org"].map(domain=>value+domain);

}

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

}

compareWithRepassword = (rule,value,callback)=>{

**const** form = **this**.props.form;

**if**(value&& **this**.state.repasswordDirty){

form.validateFields(['repassword'],{force:true});

}

callback();

}

compareWithPassword = (rule,value,callback)=>{//自定义校验器 规则 最新的值 回调

**const** form = **this**.props.form;

**if**(value && value !== form.getFieldValue('password')){

callback('密码和确认密码不一致');

}**else**{

callback();

}

}

repasswordChange = (event)=>{

**this**.setState({repasswordDirty:**this**.state.repasswordDirty||event.target.value.length>0});

}

render(){

//一旦把组件用Form.create包裹

**let** {form:{getFieldDecorator},isLogin,handleSubmit} = **this**.props;

**let** formTailItemLayout = {

wrapperCol:{offset:4,span:20}

}

**let** countrySelector = getFieldDecorator('prefix',{

initialValue:'086'

})(

<Select style={{width:70}}>

<Option value="086">086</Option>

<Option value="087">087</Option>

<Option value="088">088</Option>

</Select>

);

**let** websiteOptions = **this**.state.autoCompleteResult.map(item=>(

<AutoComplete.Option key={item}>{item}</AutoComplete.Option>

));

**let** websiteField = (

<AutoComplete onChange={this.handleWebsiteChange}>

{websiteOptions}

</AutoComplete>

)

**let** genderField = (

<Radio.Group >

<Radio value={1}>男</Radio>

<Radio value={0}>女</Radio>

</Radio.Group>

)

**let** filedItems = getFieldItems(getFieldDecorator,[

{visible:true,label:"用户名",name:"username",required:true,input:<Input/>},

{visible:true,label:"密码",name:"password",required:true,input:<Input onChange={this.repasswordChange}/>,rules:[

{validator:**this**.compareWithRepassword},

{min:1,message:'密码长度最短1位'},

{max:8,message:'密码长度最长8位'}

]},

{visible:!isLogin,label:"确认密码",name:"repassword",required:false,input:<Input/>,rules:[

{validator:**this**.compareWithPassword}

]},

{visible:!isLogin,label:"邮箱",name:"email",required:true,input:<Input/>,rules:[{type:'email',message:'必须输入一个合法的邮箱!'}]},

{visible:!isLogin,label:"性别",name:"gender",required:true,input:genderField,extra:{initialValue:1}},

{visible:!isLogin,label:"住址",name:"address",required:true,input:<Cascader options={addresses}/>},

{visible:!isLogin,label:"手机号",name:"phone",required:true,rules:[{pattern:/^1\d{10}$/,message:'请输入合法手机号'}],input:<Input addonBefore={countrySelector} style={{width:'100%'}}/>},

{visible:!isLogin,label:"网址",name:"website",input:websiteField},

{visible:!isLogin,name:"agreement",layout:formTailItemLayout,input:<Checkbox>我已经同意本协议</Checkbox>,extra:{valuePropName:'checked'}}

]);

**return** (

<FormWrapper>

<Form style={{width:'500px'}} onSubmit={handleSubmit}>

<h3>欢迎{isLogin?"登录":"注册"}</h3>

{filedItems}

<FormItem>

<Button type="primary" htmlType="submit" style={{width:'100%'}}>{isLogin?"登录":"注册"}</Button>

已有账号?<a href="#" onClick={this.props.changeLoginStatus}>立刻{isLogin?"注册":"登录"}</a>

</FormItem>

</Form>

</FormWrapper>

)

}

}

LoginForm = Form.create()(LoginForm);**const** FormWrapper = styled.div`

display:flex;

justify-content:center;

align-items:center;

height:calc(100vh - 70px );

h3{

text-align:center;

}

form{

border:1px solid #999;

border-radius:5px;

padding:20px;

}

`//一般来说在dva里，一个页面路由组件会对应一个子状态 login**export** **default** connect(

state=>state.login

)(Login);

#### **3.1.3 getFieldItems.js**

src\utils\getFieldItems.js

**import** {Form} **from** 'antd';**const** FormItem = Form.Item;**let** formItemLayout = {

labelCol:{span:4},wrapperCol:{span:20}

}**function** **getFieldItems**(getFieldDecorator,fields){

//{label,name,rules,input}

**return** fields.filter(field=>field.visible).map((field,index)=>{

**let** layout = field.layout?field.layout:formItemLayout;

field.extra = field.extra||{};

field.rules = field.rules||[];

**return** (

<FormItem key={index} label={field.label} {...layout}>

{

getFieldDecorator(field.name,{

rules:[{required:field.required,message:`${field.label}必须输入`},...field.rules],

...field.extra

})(field.input)

}

</FormItem>

)

});

}**export** **default** getFieldItems;

#### **3.1.4 addresses.js**

src\utils\addresses.js

**export** **default** [

{

value:'guangdong',

label: '广东',

children: [

{value:'guangzhou',label: '广州',},

{value:'dongguan',label: '东莞',}

]

},

{

value:'shandong',

label: '山东',

children: [

{value:'jinan',label: '济南'},

{value:'shouguang',label: '寿光',}

]

}

]

#### **3.1.4 utils/request.js**

src/utils/request.js

**import** fetch **from** 'dva/fetch';**const** BASE\_URL = 'http://127.0.0.1:7001';**export** **default** **function** (url, options={}) {

**let** token = localStorage.getItem('token');

options.headers = options.headers || {};

**if** (token) {

options.headers.authorization = token;

}

options.method = options.method || 'GET';

options.headers["Content-Type"] = "application/json";

options.headers["Accept"] = "application/json";

options.credentials = 'include';

**return** fetch(BASE\_URL + url,options).then(res => res.json());

}

### **3.3 models/login.js**

src/pages/login/models/login.js

**import** \* **as** service **from** '../services/login';**import** {decode} **from** 'jsonwebtoken';**import** {routerRedux} **from** 'dva/router';**import** {message} **from** 'antd';**export** **default** {

namespace:'login',

state:{

isLogin:true,//是否正在注册

errorInfo:'',//放着报错信息

userInfo:null//如果用户登录之后，会把用户信息放在这里

},

effects:{

\*signup({payload},{call,put}){

**let** result = **yield** call(service.signup,payload);

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

**yield** put({type:'save',payload:{isLogin:true}});

}**else**{

message.error('注册失败!');

}

},

\*login({payload},{call,put}){

**const** result = **yield** call(service.login,payload);

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

**const** userInfo = decode(result.data);

**yield** put({type:'save',payload:{userInfo}});

localStorage.setItem('token',result.data);

**yield** put(routerRedux.push('/admin'));

}**else**{

message.error('登录失败!');

}

},

\*loadUser({payload},{call,put}) {

**let** token=localStorage.getItem('token');

**if** (token) {

**const** userInfo = decode(token);

**yield** put({type:'save',payload:{userInfo}});

} **else** {

**yield** put(routerRedux.push('/login'));

}

}

},

reducers:{

save(state,action){

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

}

}

}

### **3.4 login/services/login.js**

src/pages/login/services/login.js

**import** request **from** '../../../utils/request';

//定义一个注册的方法,调用api接口**export** **function** **signup**(payload){

**return** request('/api/signup',{

method:'POST',

body:JSON.stringify(payload)

});

}//定义一个登陆的方法**export** **function** **login**(payload) {

**return** request(`/api/signin`,{

method: 'POST',

body:JSON.stringify(payload)

});

}

## **4.验证码**

### **4.1 login/index.js**

src/pages/login/index.js LoginForm

+ import getFieldItems,{formItemLayout} from '../../utils/getFieldItems';

+ const captchaUrl=`http://127.0.0.1:7001/api/captcha?ts=`;

+ refreshCaptcha = (event)=>{+ event.target.src = captchaUrl+Date.now();+ }

+ <FormItem label="验证码" {...formItemLayout} extra="证明你不是机器人">+ <Row gutter={8}>+ <Col span={12}>+ {+ getFieldDecorator('captcha',{+ rules: [{required:true,message:'必须输入验证码'}]+ })(<Input />)+ }+ </Col>+ <Col span={12}>+ <img src={captchaUrl} onClick={this.refreshCaptcha}/>+ </Col>+ </Row>+ </FormItem>

## **5.后台管理界面布局模版**

### **5.1 /admin/\_layout.js**

src/pages/admin/\_layout.js

**import** React,{Component,Fragment} **from** 'react';**import** { Layout } **from** 'antd';**const** { Header, Footer, Sider, Content } = Layout;**export** **default** **class** **Admin** **extends** **Component**{

render() {

**return** (

<Layout>

<Header>Header</Header>

<Layout>

<Sider>Sider</Sider>

<Content>{this.props.children}</Content>

</Layout>

<Footer>

珠峰培训 ©2018

</Footer>

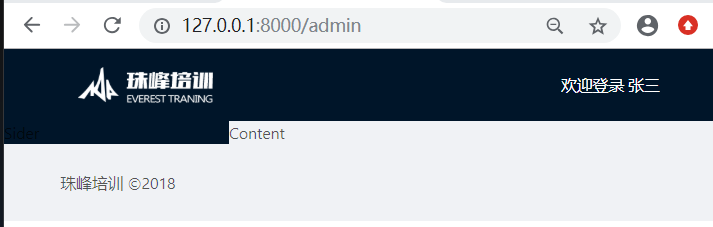
</Layout>

)

}

}

## **6. 顶部导航条**



### **6.1 components\AdminHeader\index.js**

src\components\AdminHeader\index.js

**import** React,{Component} **from** 'react';**import** {Layout} **from** 'antd';**import** {connect} **from** 'dva';**import** styles **from** './index.less';**const** {Header}=Layout;**class** **AdminHeader** **extends** **Component**{

componentWillMount() {

**this**.props.dispatch({type:'login/loadUser'});

}

render() {

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

**return** (

<Header className={styles.header}>

<img className={styles.logo} src="http://img.zhufengpeixun.cn/zfpxlogo.png" alt="logo"/>

<span className={styles.welcome}>欢迎登录 {userInfo&&userInfo.username}</span>

</Header>

)

}

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

state => state.login

)(AdminHeader);

### **6.2 components\AdminHeader\index.less**

src\components\AdminHeader\index.less

.logo{

width:120px;

height:32px;

margin:16px;

float:left;

}.welcome{

float:right;

color:#FFF;

}

### **6.3 admin/\_layout.js**

src/pages/admin/\_layout.js

**import** React,{Component} **from** 'react';**import** { Layout } **from** 'antd';**import** AdminHeader **from** '../../components/AdminHeader';**const** {Footer,Sider,Content}=Layout;**export** **default** **class** **Admin** **extends** **Component**{

render() {

**return** (

<Layout>

<AdminHeader/>

<Layout>

<Sider>Sider</Sider>

<Content>{this.props.children}</Content>

</Layout>

<Footer>

珠峰架构 ©2018

</Footer>

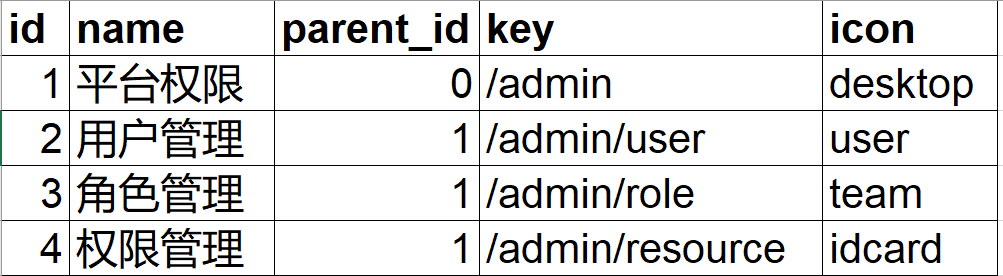
</Layout>

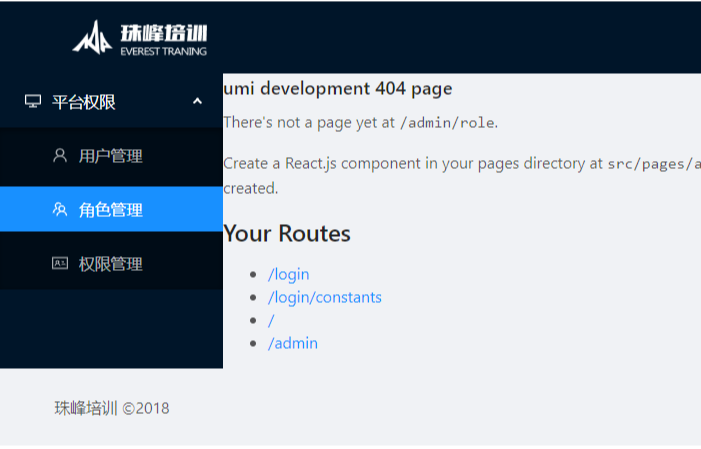
)

}

}

## **7.左侧菜单和用户页面**





### **7.1 components/MenuList.js**

src/components/MenuList.js

**import** React,{Component,Fragment} **from** 'react';**import** {Menu,Icon} **from** 'antd';**import** Link **from** 'umi/link';**import** {connect} **from** 'dva';**const** SubMenu=Menu.SubMenu;**class** **MenuList** **extends** **Component**{

renderMenus=(resources=[]) => {

**return** resources.map(resource => {

**if** (resource.children.length>0) {

**return** (

<SubMenu key={resource.key} title={<span><Icon type={resource.icon} />{resource.name}</span>}>

{this.renderMenus(resource.children)}

</SubMenu>

)

} **else** {

**return** <Menu.Item key={resource.key}><Link to={resource.key}><Icon type={resource.icon} />{resource.name}</Link></Menu.Item>;

}

});

}

render() {

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

**if** (!userInfo)

**return** null;

**return** (

<Menu

theme="dark"

defaultSelectedKeys={['/admin']}

defaultOpenKeys={['/admin']}

mode="inline"

>

{

this.renderMenus(userInfo.resources)

}

</Menu>

)

}

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

state => state.login

)(MenuList);

app\controller\user.js

**let** list = **await** app.mysql.query(`SELECT resource.\* FROM role\_user,role\_resource,resource where role\_user.role\_id = role\_resource.role\_id AND role\_resource.resource\_id = resource.id AND role\_user.user\_id = ? ORDER BY resource.id ASC`,[user.id]);

**let** resources = [];

**let** map = {};

list.forEach(item => {

item.children = [];

map[item.id] = item;

**if** (item.parent\_id == 0) {

resources.push(item);

} **else** {

map[item.parent\_id].children.push(item);

}

});

user.resources=resources;

### **7.2 admin/\_layout.js**

src/pages/admin/\_layout.js

**import** React,{Component} **from** 'react';**import** { Layout } **from** 'antd';**import** AdminHeader **from** '../../components/AdminHeader';**import** MenuList **from** '../../components/MenuList';**const** {Footer,Sider,Content}=Layout;**export** **default** **class** **Admin** **extends** **Component**{

render() {

**return** (

<Layout>

<AdminHeader/>

<Layout>

<Sider>

<MenuList/>

</Sider>

<Content>

{this.props.children}

</Content>

</Layout>

<Footer>

珠峰培训 ©2018

</Footer>

</Layout>

)

}

}

## **8. 用户管理**

### **8.1 用户列表**

#### **8.1.1 index.js**

src\pages\admin\user\index.js

**import** React,{Component,Fragment} **from** 'react';**import** {connect} **from** 'dva';**import** {Card,Table,Button,Modal,Form,Input,message,Popconfirm} **from** 'antd';**import** {PAGE\_SIZE} **from** './constants';**import** { routerRedux } **from** 'dva/router';**const** FormItem=Form.Item;

**export** **default**

@connect(

state => ({...state.user,loading:state.loading.models.user})

) **class** **user** **extends** **Component**{

render(){

**const** columns=[

{

title: '用户名',

dataIndex: 'username',

key: 'username'

},

{

title: '邮箱',

dataIndex: 'email',

key: 'email'

},

{

title: '性别',

dataIndex: 'gender',

key: 'gender',

render: (val,record)=>(

val===1?'男':'女'

)

}

]

**let** {list,loading,pageNum,total,dispatch}=**this**.props;

**const** pagination={

current: pageNum,

pageSize: PAGE\_SIZE,

showQuickJumper: true,

showTotal: (total,range) => {

**return** `共${total}条`;

},

total,

onChange: (pageNum) => {

dispatch(routerRedux.push(`/admin/user?pageNum=${pageNum}`));

}

}

**return** (

<Card>

<Table

columns={columns}

dataSource={list}

loading={loading}

rowKey={record => record.id}

pagination={pagination}

/>

</Card>

)

}

}

#### **8.1.2 constants.js**

src\pages\admin\user\constants.js

**export** **const** PAGE\_SIZE = 3;

#### **8.1.3 models\user.js**

src\pages\admin\user\models\user.js

**import** \* **as** userService **from** '../services/user';

**export** **default** {

namespace: 'user',

state: {

list: [],

pageNum:1,

total:0

},

reducers: {

save(state,{payload}) {

**return** {...state,...payload};

}

},

effects: {

\*fetch({payload: {pageNum=1}},{call,put}) {

**const** result=**yield** call(userService.fetch,pageNum);

**debugger**;

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

**let** { list,total} = result.data;

**yield** put({type:'save',payload:{list,pageNum:parseInt(pageNum),total}});

}

}

},

subscriptions: {

setup({dispatch,history}) {

**return** history.listen(({pathname,query}) => {

**if** (pathname==='/admin/user') {

dispatch({type:'fetch',payload:query});

}

});

}

}

}

#### **8.1.4 services\user.js**

src\pages\admin\user\services\user.js

**import** request **from** '@/utils/request';**import** {PAGE\_SIZE} **from** '../constants';**export** **function** **fetch**(pageNum) {

**return** request(`/api/user?pageNum=${pageNum}&pageSize=${PAGE\_SIZE}`);

}

### **8.2 新增和编辑用户**

#### **8.1.1 index.js**

src\pages\admin\user\index.js

import React,{Component,Fragment} from 'react';

import {connect} from 'dva';

import {Card,Table,Button,Modal,Form,Input,message,Popconfirm} from 'antd';

import {PAGE\_SIZE} from './constants';

import { routerRedux } from 'dva/router';

const FormItem=Form.Item;

export default

@connect(

state => ({...state.user,loading:state.loading.models.user})

)

class user extends Component{+ save = (payload) => {+ this.props.dispatch({+ type: 'user/save',+ payload+ });+ }+ onAdd=() => {+ this.save({editVisible: true,isCreate:true,record: {} });+ }+ onEditCancel=() => {+ this.save({ editVisible : false });+ }+ onEdit=(record) => {+ this.save({editVisible: true,isCreate:false,record});+ }+ onEditOk=() => {+ this.editForm.props.form.validateFields((err,values) => {+ if (err) {+ return message.error('表单校验失败!');+ } else {+ this.props.dispatch({+ type: this.props.isCreate?'user/create':'user/update',+ payload:values+ });+ }+ });+ }

render(){

const columns=[

{

title: '用户名',

dataIndex: 'username',

key: 'username'

},

{

title: '邮箱',

dataIndex: 'email',

key: 'email'

},

{

title: '性别',

dataIndex: 'gender',

key: 'gender',

render: (val,record)=>(

val===1?'男':'女'

)

},+ {+ title: '操作',+ key: 'operation',+ render: (val,record) => (+ <Fragment>+ <Button type="warning" onClick={()=>this.onEdit(record)}>编辑</Button>+ </Fragment>+ )+ }

]+ let {list,loading,pageNum,total,dispatch,isCreate,editVisible,record}=this.props;

const pagination={

current: pageNum,

pageSize: PAGE\_SIZE,

showQuickJumper: true,

showTotal: (total,range) => {

return `共${total}条`;

},

total,

onChange: (pageNum) => {

dispatch(routerRedux.push(`/admin/user?pageNum=${pageNum}`));

}

}

return (

<Card>

<Button type="warning" onClick={this.onAdd}>添加</Button>

<Table

columns={columns}

dataSource={list}

loading={loading}

rowKey={record => record.id}

pagination={pagination}

/>+ <EditModal+ wrappedComponentRef={instance =>this.editForm=instance}+ isCreate={isCreate}+ visible={editVisible}+ onOk={this.onEditOk}+ onCancel={this.onEditCancel}+ record={record}+ />

</Card>

)

}

}+@Form.create()+class EditModal extends Component{+ render() {+ let {visible,onOk,isCreate,onCancel,record,form: {getFieldDecorator}=this.props;+ let {username,email,id}=record;+ return (+ <Modal+ title={isCreate?'创建用户':'修改用户'}+ visible={visible}+ onOk={onOk}+ onCancel={onCancel}+ destroyOnClose+ >+ <Form>+ <FormItem>+ {+ getFieldDecorator('id',{+ initialValue: id+ })(<Input type="hidden" />)+ }+ </FormItem>+ <FormItem+ label="用户名"+ >+ {+ getFieldDecorator('username',{+ rules: [{+ required: true,+ message:'用户名必须输入'+ }],+ initialValue: username+ })(<Input />)+ }+ </FormItem>+ <FormItem label="邮箱" >+ {+ getFieldDecorator('email',{+ initialValue: email,+ rules: [{required: true,message:'用户名必须输入'+ }]})(<Input/>)+ }+ </FormItem>+ </Form>+ </Modal>+ )+ }+}

#### **8.1.2 models/user.js**

src/pages/admin/user/models/user.js

import \* as userService from '../services/user';

export default {

namespace: 'user',

state: {

list: [],

pageNum:1,

total:0,+ editVisible: false,+ isCreate:true,+ record: {}

},

reducers: {

save(state,{payload}) {

return {...state,...payload};

}

},

effects: {

\*fetch({payload: {pageNum=1}},{call,put}) {

const result=yield call(userService.fetch,pageNum);

debugger;

if(result.code === 0){

let { list,total} = result.data;

yield put({type:'save',payload:{list,pageNum:parseInt(pageNum),total}});

}

},+ \*create({payload},{call,put}) {+ yield call(userService.create,payload);+ yield put({type: 'fetch',payload: {pageNum:1}});+ yield put({type:'save',payload:{editVisible:false}});+ },+ \*update({payload},{call,put,select}) {+ yield call(userService.update,payload);+ let pageNum=yield select(state=>state.user.pageNum);+ yield put({type: 'fetch',payload: {pageNum}});+ yield put({type:'save',payload:{editVisible:false}});+ },

},

subscriptions: {

setup({dispatch,history}) {

return history.listen(({pathname,query}) => {

if (pathname==='/admin/user') {

dispatch({type:'fetch',payload:query});

}

});

}

}

}

#### **8.1.3 services/user.js**

src/pages/admin/user/services/user.js

import request from '@/utils/request';

import {PAGE\_SIZE} from '../constants';

export function fetch(pageNum) {

return request(`/api/user?pageNum=${pageNum}&pageSize=${PAGE\_SIZE}`);

}+export function create(values) {+ return request(`/api/user`,{+ method: 'POST',+ headers:{"Content-Type":"application/json"},+ body:JSON.stringify(values)+ });+}+export function update(values) {+ return request(`/api/user/${values.id}`,{+ method: 'PUT',+ headers:{"Content-Type":"application/json"},+ body:JSON.stringify(values)+ });+}

### **8.3 删除用户**

#### **8.3.1 index.js**

src\pages\admin\user\index.js

+ onDel=(id) => {+ this.props.dispatch({+ type: 'user/del',+ payload:id+ });+ }

render: (val,record) => (

<Fragment>

<Button type="warning" onClick={()=>this.onEdit(record) </Button>+ <Popconfirm+ okText="确认"+ cancelText="取消"+ title="确认删除此用户吗?"+ onConfirm={() => this.onDel(record.id)}>+ <Button type="danger">删除</Button>+ </Popconfirm>

</Fragment>

#### **8.3.2 models\user.js**

src\pages\admin\user\models\user.js

+ \*del({payload},{call,put}) {+ yield call(userService.del,payload);+ yield put({type: 'fetch',payload: {pageNum:1}});+ }

#### **8.3.3 services\user.js**

src\pages\admin\user\services\user.js

+ export function del(id) {+ return request(`/api/user/${id}`,{+ method: 'DELETE'+ });

### **8.4 全部删除**

#### **8.4.1 src\pages\admin\user\index.js**

src\pages\admin\user\index.js

type: 'user/del',

payload:id

});

}+ onAllDel=() => {+ this.props.dispatch({+ type: 'user/delAll',+ payload:this.props.selectedRowKeys+ });+ }

render(){

const columns=[

dispatch(routerRedux.push(`/admin/user?pageNum=${pageNum}`));

}

}+ const rowSelection={+ type: 'checkbox',+ selectedRowKeys: this.props.selectedRowKeys,+ onChange: (selectedRowKeys) => {+ this.save({selectedRowKeys});+ }

}

return (

<Card>

<Button type="warning" onClick={this.onAdd}>添加</Button>+ <Button type="danger" onClick={this.onAllDel}>全部删除</Button>

<Table

columns={columns}

dataSource={list}

loading={loading}

rowKey={record => record.id}

pagination={pagination}+ rowSelection={rowSelection}

/>

<EditModal

wrappedComponentRef={instance =>this.editForm=instance}

#### **8.4.2 user.js**

pages\admin\user\models\user.js

record: {},+ selectedRowKeys:[]+ \*delAll({payload},{call,put}) {+ yield call(userService.delAll,payload);+ yield put({type: 'fetch',payload: {pageNum:1}});+ }

#### **8.4.3 services\user.js**

src\pages\admin\user\services\user.js

+ export function delAll(ids) {+ return request(`/api/user/${ids[0]}`,{+ method: 'DELETE',+ headers: {"Content-Type": "application/json"},+ body: JSON.stringify(ids)+ });+}

### **8.5 搜索**

#### **8.5.1 src\pages\admin\user\index.js**

src\pages\admin\user\index.js

type: 'user/delAll',

payload:this.props.selectedRowKeys

});

}+ onSearch=() => {+ let values=this.searchForm.props.form.getFieldsValue();+ let where=Object.keys(values).reduce((memo,key) => {+ if (values[key]){+ memo[key]=values[key];+ }+ return memo;+ },{});+ this.props.dispatch({+ type: 'user/search',+ payload:where+ });+ }

+ let {list,loading,pageNum,total,dispatch,isCreate,editVisible,record,where}=this.props;

+ <>+ <Card>+ <SearchForm+ where={where}+ onSearch={this.onSearch}+ wrappedComponentRef={inst=>this.searchForm=inst}/>

</Card>

<Card>+ </>

+@Form.create()+class SearchForm extends Component{+ render() {+ let {form: {getFieldDecorator},onSearch,where={}}=this.props;+ return (+ <Form layout="inline">+ <FormItem+ label="用户名"+ >+ {+ getFieldDecorator('username',{initialValue:where.username})(<Input />)+ }+ </FormItem>+ <FormItem+ label="邮箱"+ >+ {+ getFieldDecorator('email',{initialValue:where.email})(<Input/>)+ }+ </FormItem>+ <FormItem>+ <Button onClick={onSearch} shape="circle" icon="search"></Button>+ </FormItem>+ </Form>+ )+ }+}

#### **8.5.2 user.js**

pages\admin\user\models\user.js

selectedRowKeys:[],+ where:{}

+ \*fetch({payload: {pageNum=1,where}},{call,put,select}) {+ if (!where) {+ where =yield select(state => state.user.where);+ }+ if (!pageNum) {+ pageNum =yield select(state => state.user.pageNum);+ }+ const result=yield call(userService.fetch,pageNum,where);+ if(result.code === 0){+ let { list,total} = result.data;+ yield put({type:'save',payload:{list,pageNum:parseInt(pageNum),total}});+ }+ },+ \*search({payload:where},{call,put}) {+ yield put({type: 'fetch',payload: {pageNum:1,where}});+ },

#### **8.5.3 services\user.js**

src\pages\admin\user\services\user.js

+import querystring from 'querystring';+export function fetch(pageNum,where) {+ let whereString=querystring.stringify(where);+ return request(`/api/user?pageNum=${pageNum}&pageSize=${PAGE\_SIZE}&${whereString}`);+}

### **8.6 点行选择行**

#### **8.6.1 src\pages\admin\user\index.js**

src\pages\admin\user\index.js

<Table

columns={columns}

dataSource={list}

loading={loading}

rowKey={record => record.id}

pagination={pagination}

rowSelection={rowSelection}+ onRow = {+ (record) => {+ return {+ onClick: () => {+ let selectedRowKeys = this.props.selectedRowKeys;+ let index = selectedRowKeys.indexOf(record.id);+ if (index == -1) {+ selectedRowKeys = [...selectedRowKeys, record.id];+ } else {+ selectedRowKeys = selectedRowKeys.filter(key => key !=record.id);+ }+ this.save({+ selectedRowKeys+ });+ }+ }+ }+ }

/>

## **9 角色管理**

### **9.1 role\index.js**

src\pages\admin\role\index.js

**import** React,{Component,Fragment} **from** 'react';**import** {connect} **from** 'dva';**import** {Card,Table,Button,Modal,Form,Input,message,Popconfirm} **from** 'antd';**import** {PAGE\_SIZE} **from** './constants';**import** { routerRedux } **from** 'dva/router';**const** FormItem=Form.Item;**const** ENTITY='role';**export** **default**

@connect(

state => ({...state[ENTITY],loading:state.loading.models[ENTITY]})

) **class** **Role** **extends** **Component**{

save = (payload) => {

**this**.props.dispatch({

type: `${ENTITY}/save`,

payload

});

}

onAdd=() => {

**this**.save({editVisible: true,isCreate:true,record: {} });

}

onEditCancel=() => {

**this**.save({ editVisible : false });

}

onEditOk=() => {

**this**.editForm.props.form.validateFields((err,values) => {

**if** (err) {

**return** message.error('表单校验失败!');

} **else** {

**this**.props.dispatch({

type: **this**.props.isCreate?`${ENTITY}/create`:`${ENTITY}/update`,

payload:values

});

}

});

}

onEdit=(record) => {

**this**.save({editVisible: true,isCreate:false,record});

}

onDel=(id) => {

**this**.props.dispatch({

type: `${ENTITY}/del`,

payload:id

});

}

onAllDel=() => {

**this**.props.dispatch({

type: `${ENTITY}/delAll`,

payload:**this**.props.selectedRowKeys

});

}

onSearch=() => {

**let** values=**this**.searchForm.props.form.getFieldsValue();

**let** where=Object.keys(values).reduce((memo,key) => {

**if** (values[key]){

memo[key]=values[key];

}

**return** memo;

},{});

**this**.props.dispatch({

type: `${ENTITY}/search`,

payload:where

});

}

render(){

**const** columns=[

{

title: '名称',

dataIndex: 'name',

key: 'name'

},

{

title: '操作',

key: 'operation',

render: (val,record) => (

<Fragment>

<Button type="warning" onClick={()=>this.onEdit(record)}>编辑</Button>

<Popconfirm

okText="确认"

cancelText="取消"

title="确认删除此用户吗?"

onConfirm={() => this.onDel(record.id)}>

<Button type="danger">删除</Button>

</Popconfirm>

</Fragment>

)

}

]

**let** {list,loading,pageNum,total,dispatch,isCreate,editVisible,record,where}=**this**.props;

**const** pagination={

current: pageNum,

pageSize: PAGE\_SIZE,

showQuickJumper: true,

showTotal: (total,range) => {

**return** `共${total}条`;

},

total,

onChange: (pageNum) => {

dispatch(routerRedux.push(`/admin/${ENTITY}?pageNum=${pageNum}`));

}

}

**const** rowSelection={

type: 'checkbox',

selectedRowKeys: **this**.props.selectedRowKeys,

onChange: (selectedRowKeys) => {

**this**.save({selectedRowKeys});

}

}

**return** (

<>

<Card>

<SearchForm

where={where}

onSearch={this.onSearch}

wrappedComponentRef={inst=>this.searchForm=inst}/>

</Card>

<Card>

<Button type="warning" onClick={this.onAdd}>添加</Button>

<Button type="danger" onClick={this.onAllDel}>全部删除</Button>

<Table

columns={columns}

dataSource={list}

loading={loading}

rowKey={record => record.id}

pagination={pagination}

rowSelection={rowSelection}

onRow = {

(record) => {

return {

onClick: () => {

let selectedRowKeys = this.props.selectedRowKeys;

let index = selectedRowKeys.indexOf(record.id);

if (index === -1) {

selectedRowKeys = [...selectedRowKeys, record.id];

} else {

selectedRowKeys = selectedRowKeys.filter(key => key !=record.id);

}

this.save({

selectedRowKeys

});

}

}

}

}

/>

<EditModal

wrappedComponentRef={instance =>this.editForm=instance}

isCreate={isCreate}

visible={editVisible}

onOk={this.onEditOk}

onCancel={this.onEditCancel}

record={record}

/>

</Card>

</>

)

}

}

@Form.create()**class** **SearchForm** **extends** **Component**{

render() {

**let** {form: {getFieldDecorator},onSearch,where={}}=**this**.props;

**return** (

<Form layout="inline">

<FormItem

label="角色名称"

>

{

getFieldDecorator('name',{initialValue:where.name})(<Input />)

}

</FormItem>

<FormItem>

<Button onClick={onSearch} shape="circle" icon="search"></Button>

</FormItem>

</Form>

)

}

}

@Form.create()**class** **EditModal** **extends** **Component**{

render() {

**let** {visible,onOk,isCreate,onCancel,record,form: {getFieldDecorator}}=**this**.props;

**let** {id,name}=record;

**return** (

<Modal

title={isCreate?'创建角色':'修改角色'}

visible={visible}

onOk={onOk}

onCancel={onCancel}

destroyOnClose

>

<Form>

<FormItem>

{

getFieldDecorator('id',{

initialValue: id

})(<Input type="hidden" />)

}

</FormItem>

<FormItem

label="用户名"

>

{

getFieldDecorator('name',{

rules: [{

required: true,

message:'名称必须输入'

}],

initialValue: name

})(<Input />)

}

</FormItem>

</Form>

</Modal>

)

}

}

### **9.2 role\constants.js**

src\pages\admin\role\constants.js

**export** **const** PAGE\_SIZE = 3;

### **9.3 models\role.js**

src\pages\admin\role\models\role.js

**import** \* **as** service **from** '../services/role';**const** ENTITY='role';**export** **default** {

namespace: ENTITY,

state: {

list: [],

pageNum:1,

total:0,

editVisible: false,

isCreate:true,

record: {},

selectedRowKeys:[],

where:{}

},

reducers: {

save(state,{payload}) {

**return** {...state,...payload};

}

},

effects: {

\*fetch({payload: {pageNum=1,where}},{call,put,select}) {

**if** (!where) {

where =**yield** select(state => state[ENTITY].where);

}

**if** (!pageNum) {

pageNum =**yield** select(state => state[ENTITY].pageNum);

}

**const** result=**yield** call(service.fetch,pageNum,where);

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

**let** { list,total} = result.data;

**yield** put({type:'save',payload:{list,pageNum:parseInt(pageNum),total}});

}

},

\*search({payload:where},{call,put}) {

**yield** put({type: 'fetch',payload: {pageNum:1,where}});

},

\*create({payload},{call,put}) {

**yield** call(service.create,payload);

**yield** put({type: 'fetch',payload: {pageNum:1}});

**yield** put({type:'save',payload:{editVisible:false}});

},

\*update({payload},{call,put,select}) {

**yield** call(service.update,payload);

**let** pageNum=**yield** select(state=>state[ENTITY].pageNum);

**yield** put({type: 'fetch',payload: {pageNum}});

**yield** put({type:'save',payload:{editVisible:false}});

},

\*del({payload},{call,put}) {

**yield** call(service.del,payload);

**yield** put({type: 'fetch',payload: {pageNum:1}});

},

\*delAll({payload},{call,put}) {

**yield** call(service.delAll,payload);

**yield** put({type: 'fetch',payload: {pageNum:1}});

}

},

subscriptions: {

setup({dispatch,history}) {

**return** history.listen(({pathname,query}) => {

**if** (pathname===`/admin/${ENTITY}`) {

dispatch({type:'fetch',payload:query});

}

});

}

}

}

### **9.4 services\role.js**

src\pages\admin\role\services\role.js

**import** request **from** '@/utils/request';**import** {PAGE\_SIZE} **from** '../constants';**import** querystring **from** 'querystring';**const** ENTITY='role';**export** **function** **fetch**(pageNum,where) {

**let** whereString=querystring.stringify(where);

**return** request(`/api/${ENTITY}?pageNum=${pageNum}&pageSize=${PAGE\_SIZE}&${whereString}`);

}**export** **function** **create**(values) {

**return** request(`/api/${ENTITY}`,{

method: 'POST',

headers:{"Content-Type":"application/json"},

body:JSON.stringify(values)

});

}**export** **function** **update**(values) {

**return** request(`/api/${ENTITY}/${values.id}`,{

method: 'PUT',

headers:{"Content-Type":"application/json"},

body:JSON.stringify(values)

});

}**export** **function** **del**(id) {

**return** request(`/api/${ENTITY}/${id}`,{

method: 'DELETE'

});

}**export** **function** **delAll**(ids) {

**return** request(`/api/${ENTITY}/${ids[0]}`,{

method: 'DELETE',

headers: {"Content-Type": "application/json"},

body: JSON.stringify(ids)

});

}

## **10. 设置权限**

### **10.1 service\role.js**

//list resourceIds=代表这个角色所拥有的资源ID数组**for** (**let** i = 0; i < list.length; i++) {

**let** rows = **await** app.mysql.select('role\_resource', {

where: { role\_id: list[i].id }

});//[{role\_id:1,resource\_id:2},{role\_id:1,resource\_id:3}]

list[i].resourceIds = rows.map(item => item.resource\_id);//[2,3]

rows = **await** app.mysql.select('role\_user', {

where: { role\_id: list[i].id }

});//[{role\_id:1,resource\_id:2},{role\_id:1,resource\_id:3}]

list[i].userIds = rows.map(item => item.user\_id);//[2,3]

}

### **10.2 role/index.js**

+ import {Tree,Card,Table,Button,Modal,Form,Input,message,Popconfirm} from 'antd';+const TreeNode=Tree.TreeNode;

+ setRolePermission=() => {+ if (this.props.selectedRows.length===1) {+ let record=this.props.selectedRows[0];+ this.save({setPermissionVisible:true,record,checkedKeys:record.resourceIds});+ } else {+ message.error('为角色设置权限时要选择并且只能选择一个角色!');+ }+ }+ onCheckPermission=(checkedKeys) => {+ this.save({checkedKeys});+ }+ setRolePermissionOk=() => {+ this.props.dispatch({+ type: `${ENTITY}/setRolePermission`+ });+ }

+ let {list,loading,pageNum,total,dispatch,isCreate,editVisible,record,where,setPermissionVisible,resources,checkedKeys}=this.props;

+const rowSelection={+ type: 'checkbox',+ selectedRowKeys: this.props.selectedRowKeys,+ selectedRows: this.props.selectedRows,+ onChange: (selectedRowKeys,selectedRows) => {+ this.save({selectedRowKeys,selectedRows});+ }+}

+ <Button icon="solution" onClick={this.setRolePermission}>设置权限</Button>

<Table+ rowSelection={rowSelection}

+ <PermissonModal+ visible={setPermissionVisible}+ record={record}+ resources={resources}+ checkedKeys={checkedKeys}+ onCheck={this.onCheckPermission}+ onOk={this.setRolePermissionOk}+ onCancel={()=>this.save({setPermissionVisible:false})}+ />

+class PermissonModal extends Component{+ renderResources=(children) => {+ return children.map(child => {+ if (child.children.length > 0) {+ return (+ <TreeNode title={child.name} key={child.id}>+ {this.renderResources(child.children)}+ </TreeNode>+ )+ } else {+ return <TreeNode title={child.name} key={child.id}/>+ }+ });+ }+ render() {+ let {onCheck,visible,onOk,onCancel,checkedKeys,resources=[]}=this.props;+ return (+ <Modal+ visible={visible}+ title="为角色设置权限"+ onOk={onOk}+ onCancel={onCancel}+ destroyOnClose+ >+ <Tree+ checkable+ defaultExpandAll+ onCheck={onCheck}+ checkedKeys={checkedKeys}+ >+ <TreeNode title="所有权限" key={0} disabled>+ {this.renderResources(resources)}+ </TreeNode>+ </Tree>+ </Modal>+ )+ }+}

### **10.3 models\role.js**

src\pages\admin\role\models\role.js

+ selectedRows: [],+ checkedKeys: [],+ resources:[],+ setPermissionVisible:false

+ \*getResource({payload},{call,put}) {+ const resources=yield call(service.getResources);+ yield put({type:'save',payload:{resources}});+ },+ \*setRolePermission({payload},{call,put,select}) {+ let {record,checkedKeys}=yield select(state => state[ENTITY]);+ yield call(service.setRolePermission,{roleId:record.id,resourceIds:checkedKeys});+ yield put({type: 'fetch',payload: {}});+ yield put({type:'save',payload:{setPermissionVisible:false,selectedRowKeys:[],selectedRows:[]}});+ }

if (pathname===`/admin/${ENTITY}`) {

dispatch({type:'fetch',payload:query});+ dispatch({type:'getResource'});

}

### **10.4 role.js**

src\pages\admin\role\services\role.js

+ export function getResources() {+ return request(`/api/role/getResource`);+}+export function setRolePermission(values) {+ return request(`/api/role/setResource`,{+ method: 'POST',+ headers:{"Content-Type":"application/json"},+ body:JSON.stringify(values)+ });+}

## **11.给角色加用户**

### **11.1 roles/index.js**

+ import { Transfer} from 'antd';+ setUser = () => {+ let selectedRows = this.props.selectedRows;+ if (selectedRows.length == 1) {+ let record = selectedRows[0];+ this.save({ setUserVisible: true, record, targetKeys: record.userIds });+ } else {+ message.warn('请选中并且只能选中一个角色!');+ }+ }+ setUserOk = () => {+ this.props.dispatch({+ type: `${ENTITY}/setUser`+ });+ }+ onSetUserChange = (targetKeys) => {+ this.save({ targetKeys });+ }

+ let {list,loading,pageNum,total,dispatch,isCreate,editVisible,record,

where,setPermissionVisible,resources,checkedKeys, setUserVisible, targetKeys, users}=this.props;+ <Button style={{ marginLeft: 8 }} type="primary" icon="plus-circle" onClick={this.setUser}>分配用户</Button>

+ let selectedRowKeys=this.props.selectedRowKeys;+ let selectedRows = this.props.selectedRows;+ let index = selectedRowKeys.indexOf(record.id);+ if (index===-1) {+ selectedRowKeys=[...selectedRowKeys,record.id];+ selectedRows=[...selectedRows,record];+ } else {+ selectedRowKeys=selectedRowKeys.filter(key => key!=record.id);+ selectedRows=selectedRows.filter(row => row.id!=record.id);+ }+ this.save({selectedRowKeys,selectedRows});

+ <SetUserModal+ visible={setUserVisible}+ record={record}+ onOk={this.setUserOk}+ targetKeys={targetKeys}+ onChange={this.onSetUserChange}+ users={users}+ onCancel={() => this.save({ setUserVisible: false })}+/>

+ class SetUserModal extends Component {+ render() {+ let { visible, onOk, onCancel, record, targetKeys, onChange, users } = this.props;+ return (+ <Modal+ title={`为 ${record.name} 分配用户`}+ visible={visible}+ onOk={onOk}+ okText={"确定"}+ cancelText={"取消"}+ onCancel={onCancel}+ destroyOnClose+ >+ <Transfer+ dataSource={users}+ targetKeys={targetKeys}+ titles={["待选用户", "已选用户"]}+ onChange={onChange}+ render={row => row.username}+ rowKey={row => row.id}+ />+ </Modal>+ )+ }+}

### **11.2 models/roles.js**

src/pages/admin/roles/models/roles.js

+ setPermissionVisible:false,+ setUserVisible: false,//窗口是否显示+ targetKeys: [],//选中的用户+ users: [] //所有的用户

+ \*getUser({ }, { call, put }) {+ let users = yield call(service.getUser);+ yield put({ type: 'save', payload: { users } });+ },+ \*setUser({ }, { call, put, select }) {+ let record = yield select(state => state.role.record);+ let targetKeys = yield select(state => state.role.targetKeys);+ yield call(service.setUser, {+ roleId: record.id,+ userIds: targetKeys+ });+ yield put({ type: 'save', payload: { setUserVisible: false, selectedRowKeys: [], selectedRows: [] } });+ yield put({ type: 'fetch', payload: {} });+ }

dispatch({type:'getResource'});+ dispatch({ type: 'getUser' });

### **11.3 services/roles.js**

src/pages/admin/roles/services/roles.js

+export function getUser() {+ return request(`/api/role/getUser`);+}+export function setUser(values) {+ return request(`/api/role/setUser`, {+ method: 'POST',+ headers: {+ "Content-Type": "application/json"+ },+ body: JSON.stringify(values)+ });