link: null
title: 珠峰架构师成长计划
description: .webpackrc.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=187 sentences=705, words=4672

1. 初始化项目

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

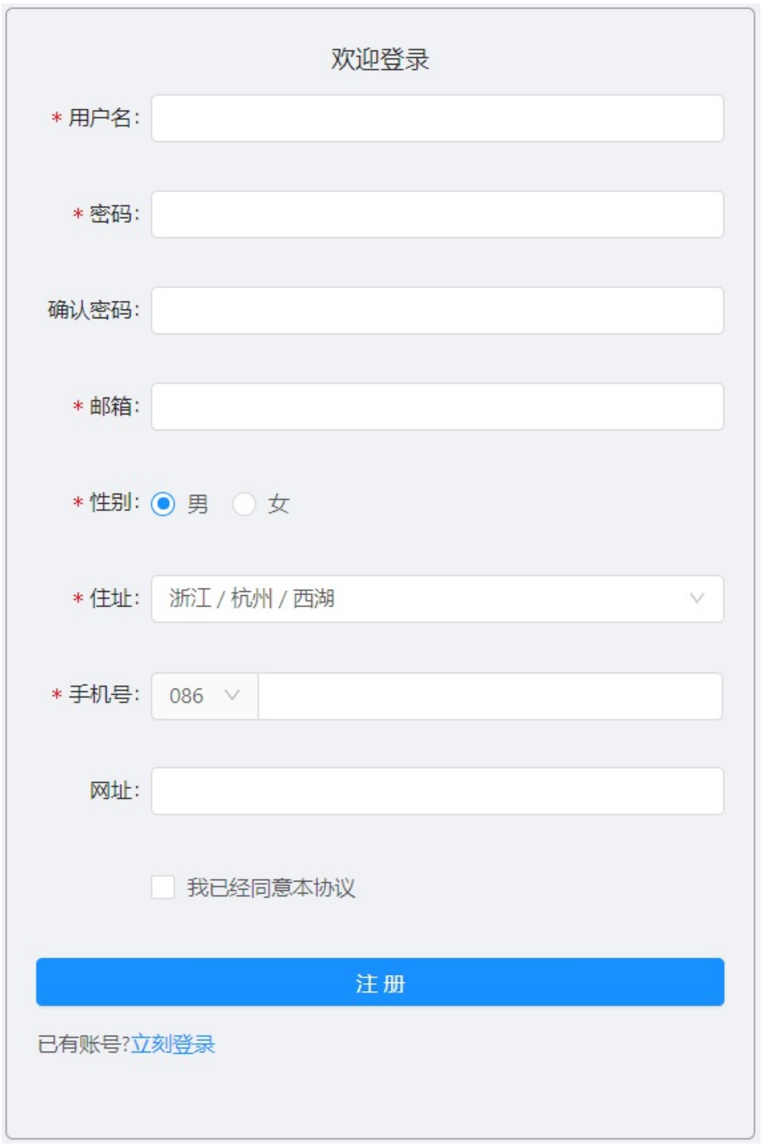
```
$ mkdir myapp && cd myapp
$ yarn create umi
$ cnpm i styled-components jsonwebtoken -S
$ npm start
```

2. 配置别名

.webpackrc.js

```
const {resolve}=require('path');
export default {
    "alias":{
        "@":resolve("src")
    }
}
```

3. 注册登录



欢迎登录

* 用户名:

* 密码:

确认密码:

* 邮箱:

* 性别: ● 男 ○ 女

* 住址: 浙江 / 杭州 / 西湖

* 手机号: 086

网址:

☐ 我已经同意本协议

注 册

已有账号?立刻登录

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';
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)=>{
      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)=>{
       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(){

        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">086Option>
                <Option value="087">087Option>
                <Option value="088">088Option>
            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;
  }
`

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){

   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,
      errorInfo:'',
      userInfo:null
  },
  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';

export function signup(payload){
  return request('/api/signup',{
      method:'POST',
      body:JSON.stringify(payload)
  });
}

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();
+ }


+
+
+
+               {
+                   getFieldDecorator('captcha',{
+                       rules: [{required:true,message:'必须输入验证码'}]
+                   })()
+               }
+
+
+
+
+
+
```

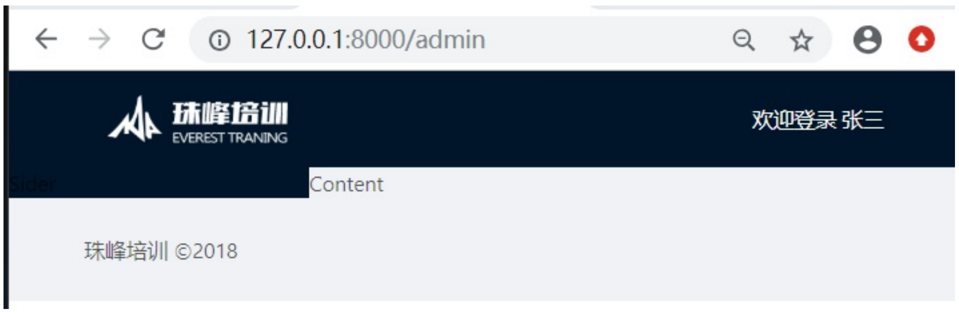
5.后台管理界面布局模版

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>HeaderHeader>
                <Layout>
                    <Sider>SiderSider>
                    <Content>{this.props.children}Content>
                Layout>
                <Footer>
                   珠峰培训 ©2018
                Footer>
            Layout>
        )
    }
}
```

6. 顶部导航条


127.0.0.1:8000/admin

欢迎登录 张三

Sider

Content

珠峰培训 ©2018

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);
```

src\components\AdminHeader\index.less

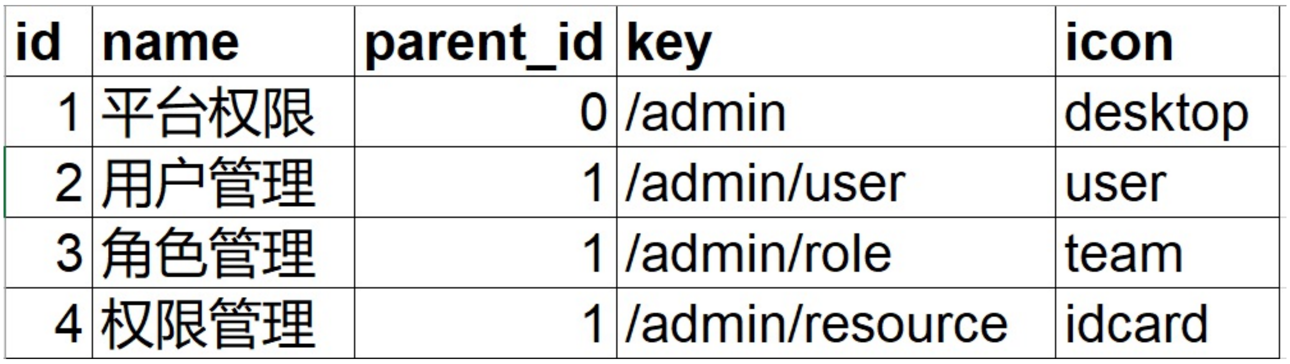
```
.logo{
    width:120px;
    height:32px;
    margin:16px;
    float:left;
}
.welcome{
    float:right;
    color:#FFF;
}
```

6.3 admin/_layout.js

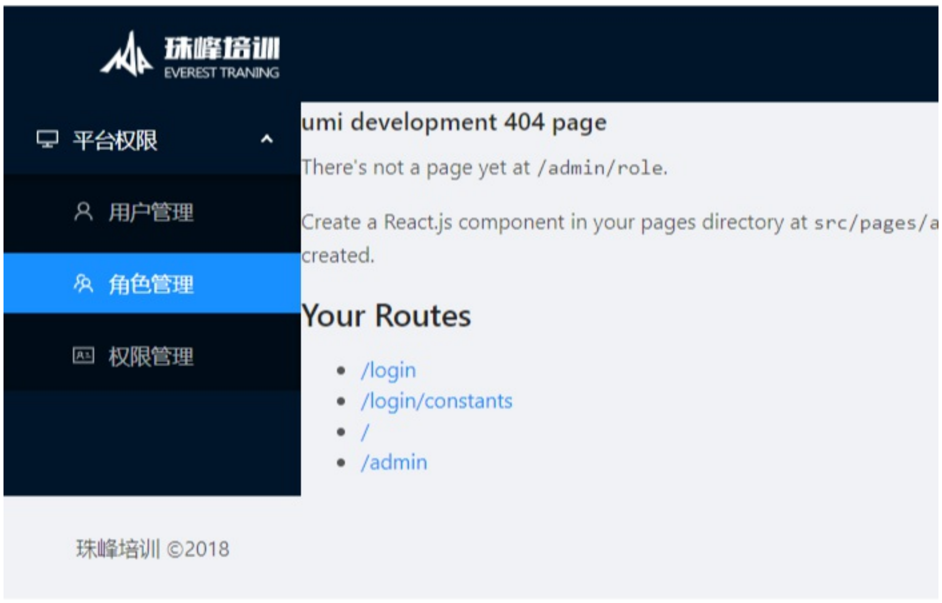
src/pages/admin/_layout.js

```
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>SiderSider>
                    <Content>{this.props.children}Content>
                Layout>
                <Footer>
                   珠峰架构 ©2018
                Footer>
            Layout>
        )
    }
}
```

7.左侧菜单和用户页面



id	name	parent_id	key	icon
1	平台权限	0	/admin	desktop
2	用户管理	1	/admin/user	user
3	角色管理	1	/admin/role	team
4	权限管理	1	/admin/resource	idcard



umi development 404 page

There's not a page yet at /admin/role.

Create a React.js component in your pages directory at src/pages/a
created.

Your Routes

- /login
- /login/constants
- /
- /admin

珠峰培训 ©2018

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',
                data
                key: 'username'
            },
            {
                title: '邮箱',
                data
                key: 'email'
            },
            {
                title: '性别',
                data
                key: 'gender',
                render: (val,record)=>(
                    val
                )
            },
+            {
+                title: '操作',
+                key: 'operation',
+                render: (val,record) => (
+
+                        this.onEdit(record)}>编辑
+
+                )
+            }
        ]
+        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 (

                添加
                 record.id}
                        pagination={pagination}
                />
+
+                    wrappedComponentRef={instance =>this.editForm=instance}
+                    isCreate={isCreate}
+                    visible={editVisible}
+                    onOk={this.onEditOk}
+                    onCancel={this.onEditCancel}
+                    record={record}
+                />

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

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
               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
                    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) => (

           this.onEdit(record)
+
+               okText="确认"
+               cancelText="取消"
+               title="确认删除此用户吗?"
+               onConfirm={() => this.onDel(record.id)}>
+                 删除
+

```

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 (

                添加
+                 全部删除
                 record.id}
                        pagination={pagination}
+                       rowSelection={rowSelection}
                />
                 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;

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

+    </>

+@Form.create()
+class SearchForm extends Component{
+    render() {
+        let {form: {getFieldDecorator},onSearch,where={}}=this.props;
+        return (
+
+
+                        label="用户名"
+                    >
+                        {
+                            getFieldDecorator('username',{initialValue:where.username})()
+                        }
+
+
+                        label="邮箱"
+                    >
+                        {
+                            getFieldDecorator('email',{initialValue:where.email})()
+                        }
+
+
+
+
+
+        )
+    }
+}
```

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

```
  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

```

for (let i = 0; i < list.length; i++) {
    let rows = await app.mysql.select('role_resource', {
        where: { role_id: list[i].id }
    });
    list[i].resourceIds = rows.map(item => item.resource_id);
    rows = await app.mysql.select('role_user', {
         where: { role_id: list[i].id }
    });
   list[i].userIds = rows.map(item => item.user_id);
}
```

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});
+           }
+}

+ 设置权限
+    rowSelection={rowSelection}

+
+                    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 (
+
+                        {this.renderResources(child.children)}
+
+                )
+            } else {
+                return
+            }
+        });
+    }
+    render() {
+        let {onCheck,visible,onOk,onCancel,checkedKeys,resources=[]}=this.props;
+        return (
+
+                visible={visible}
+                title="为角色设置权限"
+                onOk={onOk}
+                onCancel={onCancel}
+                destroyOnClose
+            >
+
+                    checkable
+                    defaultExpandAll
+                    onCheck={onCheck}
+                    checkedKeys={checkedKeys}
+                >
+
+                        {this.renderResources(resources)}
+
+
+
+        )
+    }
+}
```

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
                    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;
+                 分配用户

+                              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});

+
+                         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 (
+
+                title={`为 ${record.name} 分配用户`}
+                visible={visible}
+                onOk={onOk}
+                okText={"确定"}
+                cancelText={"取消"}
+                onCancel={onCancel}
+                destroyOnClose
+            >
+
+                    dataSource={users}
+                    targetKeys={targetKeys}
+                    titles={["待选用户", "已选用户"]}
+                    onChange={onChange}
+                    render={row => row.username}
+                    rowKey={row => row.id}
+                />
+
+        )
+    }
+}
```

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)
+    });
```