link: null
title: 珠峰架构师成长计划
description: src/pages/login/index.js
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=73 sentences=326, words=2288

1. 初始化项目

```
dva new chat
cd chat
npm i
npm start
cnpm i styled-components -S
```

2.绘制登录表单

src/pages/login/index.js

```
import React,{Component} from 'react';
import styled from 'styled-components';
import {connect} from 'dva';
import {Form,Input,message,Button} from 'antd';
class Login extends Component{
    handleSubmit=(event) => {
        event.preventDefault();
        this.loginForm.props.form.validateFields((err,values) => {
            if (err) {
                return message.warn('登录信息输入不合法!请重新输入!');
            } else {
                this.props.dispatch({type:'login/login'});
            }
        });
    }
    render() {
        return (
            <WrappedLayout>
                <WrappedLoginForm
                    handleSubmit={this.handleSubmit}
                    wrappedComponentRef={inst=>this.loginForm=inst}
                />
            WrappedLayout>
       )
   }
}
class LoginForm extends Component{
    render() {
        const {form: {getFieldDecorator},handleSubmit}=this.props;
        return (
            <Form onSubmit={handleSubmit}>
                <h4>请输入邮箱登录h4>
                <Form.Item>
                    {
                      getFieldDecorator('email',{
                        rules: [
                            {required:true,message:'请输入邮箱地址'},
                            {type:'email',message:'请输入合法的邮箱地址'}
                        ]
                      })(<Input placeholder="请输入邮箱"/>)
                    }
                Form.Item>
                <Form.Item>
                    <Button htmlType="submit" type="primary">登录Button>
                Form.Item>
            Form>
        )
    }
}
const WrappedLoginForm =  Form.create()(LoginForm);
const WrappedLayout=styled.div`
  width:100%;
  height:100%;
  display:flex;
  justify-content:center;
  align-items:center;
  form {
      width:300px;
      margin-top:-500px;
      padding:20px;
      border:1px solid #999;
      border-radius:5px;
      box-shadow:1px 1px 2px 1px,-1px -1px 2px 1px;
      h4{
          text-align:center;
          font-weight:normal;
      }
      button{
          width:100%;
      }
  }
`

export default connect(
    state=>state.login
)(Login);
```

3. 实现登录

src/pages/login/index.js

```
import React,{Component} from 'react';
import styled from 'styled-components';
import {connect} from 'dva';
import {Form,Input,message,Button} from 'antd';
class Login extends Component{
    handleSubmit=(event) => {
        event.preventDefault();
        this.loginForm.props.form.validateFields((err,values) => {
            if (err) {
                return message.warn('登录信息输入不合法!请重新输入!');
            } else {
                this.props.dispatch({type:'login/login',payload:values});
            }
        });
    }
    render() {
        return (
            <WrappedLayout>
                <WrappedLoginForm
                    handleSubmit={this.handleSubmit}
                    wrappedComponentRef={inst=>this.loginForm=inst}
                />
            WrappedLayout>
       )
   }
}
class LoginForm extends Component{
    render() {
        const {form: {getFieldDecorator},handleSubmit}=this.props;
        return (
            <Form onSubmit={handleSubmit}>
                <h4>请输入邮箱登录h4>
                <Form.Item>
                    {
                      getFieldDecorator('email',{
                        rules: [
                            {required:true,message:'请输入邮箱地址'},
                            {type:'email',message:'请输入合法的邮箱地址'}
                        ]
                      })(<Input placeholder="请输入邮箱"/>)
                    }
                Form.Item>
                <Form.Item>
                    <Button htmlType="submit" type="primary">登录Button>
                Form.Item>
            Form>
        )
    }
}
const WrappedLoginForm =  Form.create()(LoginForm);
const WrappedLayout=styled.div`
  width:100%;
  height:100%;
  display:flex;
  justify-content:center;
  align-items:center;
  form {
      width:300px;
      padding:20px;
      border:1px solid #999;
      border-radius:5px;
      box-shadow:1px 1px 2px 1px,-1px -1px 2px 1px;
      h4{
          text-align:center;
          font-weight:normal;
      }
      button{
          width:100%;
      }
  }
`

export default connect(
    state=>state.login
)(Login);
```

src/pages/login/models/login.js

```
import * as service from '../services/login';
import {routerRedux} from 'dva/router';
export default {

    namespace: 'login',

    state: {
        user: null,
        error:null
    },

    subscriptions: {
      setup({ dispatch, history }) {
      },
    },

    effects: {
        *login({payload},{call,put}) {
            const {data,error,code}=yield call(service.login,payload);
             if (code == 0) {
                yield put({type:'setUser',payload:data});
                yield put(routerRedux.push('/rooms'));
            } else {
                yield put({type:'setError',payload:error});
            }
      }
    },

    reducers: {
        setUser(state,{payload}) {
          return { ...state, user:payload };
        },
        setError(state,{payload}) {
          return { ...state, error:payload };
      }
    },
};
```

src/pages/login/services/login.js

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

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

src/utils/request.js

```
const BaseUrl='http://localhost:7001';
export default function request(url, options) {
  return fetch(BaseUrl+url, options)
    .then(checkStatus)
    .then(parseJSON)
    .catch(err => ({ err }));
}
```

4. 实现房间列表功能

pages/rooms/index.js

```
import React,{Component} from 'react';
import styled from 'styled-components';
import {connect} from 'dva';
import {Form,Input,message,Button,Layout,Menu,Breadcrumb,Card,Row,Col} from 'antd';
import logo from '../../assets/logo.png';
import styles from './index.less';
import Link from 'umi/link';
const {Header,Content,Footer}=Layout;
class Rooms extends Component{
    handleSubmit=(event) => {
        event.preventDefault();
        this.loginForm.props.form.validateFields((err,values) => {
            if (err) {
                return message.warn('登录信息输入不合法!请重新输入!');
            } else {
                this.props.dispatch({type:'login/login',payload:values});
            }
        });
    }
    handleKeywordChange=(event) => {
        this.props.dispatch({type: 'rooms/setKeyword',payload: event.target.value});
    }
    createRoom=() => {
        this.props.dispatch({type: 'rooms/createRoom'});
    }
    render() {
        const {list=[]}=this.props;
        return (
            <Layout>
                <Header>
                    <img src={logo} className={styles.logo}/>
                    <Menu
                        theme="dark"
                        mode="horizontal"
                        defaultSelectedKeys={['/rooms']}
                        style={{lineHeight:'64px'}}
                    >
                        <Menu.Item key="/rooms">房间列表Menu.Item>
                    Menu>
                Header>
                <Content style={{padding:'0 50px'}}>
```

```
                    <Breadcrumb style={{margin: '16px 0'}}>
                        <Breadcrumb.Item>房间列表Breadcrumb.Item>
                    Breadcrumb>
                    <SearchForm
                          createRoom={this.createRoom}
                          handleKeywordChange={this.handleKeywordChange}
                          size={list.length}
                        />
                    <Card>
                        <Row  gutter={16}>
                            {
                                list.map(room => (
                                    <Col span={6} style={{padding:'5px 0'}}>
                                        <Link to={`/rooms/${room._id}`}>
                                            <Card
                                                hoverable
                                                style={{ width: 240 }}
                                                cover={<img alt="example" src="https://os.alipayobjects.com/rmsportal/QBnOOoLaAfKPirc.png" />}
                                            >
                                                <Card.Meta
                                                    title={room.name}
                                                />
                                            Card>
                                        Link>
                                    Col>
                                ))
                            }
                        Row>
                    Card>
                Content>
                <Footer style={{textAlign:'center'}}>
                    珠峰培训 @2018
                Footer>
            Layout>
        )
    }
}
class SearchForm extends Component{
    render() {
        const {handleKeywordChange,createRoom,size}=this.props;
        const formItemLayout={
            labelCol: {span: 8},
            wrapperCol:{span:16}
        }
        return (
            <Form>
                <Form.Item label="请输入房间名关键字" {...formItemLayout}>
                    <Row gutter={18}>
                        <Col span={12}><Input placeholder="请输入关键字" onChange={handleKeywordChange} />Col>
                        <Col span={12}>
                            {size==0&&<Button onClick={createRoom}>创建房间Button>}

                        Col>
                    Row>
                Form.Item>
            Form>
        )
    }
}

export default connect(
    state => ({
        list: state.rooms.list.filter(item => {
            return item.name&&item.name.indexOf(state.rooms.keyword)!=-1;
        })
    })
)(Rooms);
```

pages/rooms/models/rooms.js

```
import * as service from '../services/rooms';
import {routerRedux} from 'dva/router';
export default {

    namespace: 'rooms',

    state: {
        list: [],
        error: null,
        keyword:''
    },

    subscriptions: {
        setup({dispatch,history}) {
            history.listen(({pathname,query}) => {
                if (pathname == '/rooms') {
                    dispatch({type:'list'});
                }
            });
      },
    },

    effects: {
        *list({payload},{call,put}) {
            const {data,error,code}=yield call(service.list,payload);
             if (code == 0) {
                yield put({type:'listed',payload:data});
            } else {
                yield put({type:'setError',payload:error});
            }
        },
        *createRoom({},{call,put,select}) {
            let keyword=yield select(state => state.rooms.keyword);
            const {data,error,code}=yield call(service.createRoom,{name:keyword});
             if (code == 0) {
                yield put({type:'list'});
            } else {
                yield put({type:'setError',payload:error});
            }
      }
    },

    reducers: {
        listed(state,{payload}) {
          return { ...state, list:payload };
        },
        setError(state,{payload}) {
          return { ...state, error:payload };
        },
        setKeyword(state,{payload}) {
            return {...state,keyword:payload};
        }
    },
};
```

services/rooms.js

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

export function list(keyword='') {
    return request(`/rooms`,{
        method: 'GET'
    });
}

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

5. 聊天页面

src/pages/login/models/login.js

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

    namespace: 'login',

    state: {
        user: null,
        token:null,
        error:null
    },

    subscriptions: {
      setup({ dispatch, history }) {
      },
    },

    effects: {
        *login({payload},{call,put}) {
            const {data:token,error,code}=yield call(service.login,payload);
            if (code==0) {
                let user=decode(token);
                localStorage.setItem('token',token);
                yield put({type:'setUser',payload:user});
                yield put(routerRedux.push('/rooms'));
            } else {
                yield put({type:'setError',payload:error});
            }
        },
        *validate({payload},{call,put}) {
            let token=localStorage.getItem('token')||'';
            if (token) {
                let user=decode(token);
                yield put({type:'setUser',payload:user});
            }else{
                yield put(routerRedux.push('/login'));
            }
        },
        *logout({payload},{call,put}) {
            yield put({type: 'setUser',payload: null});
            localStorage.removeItem('token');
            yield put(routerRedux.push('/login'));
        }
    },

    reducers: {
        setUser(state,{payload}) {
          return { ...state, user:payload };
        },
        setError(state,{payload}) {
          return { ...state, error:payload };
      }
    },
};
```

src/pages/login/services/login.js

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

export function login(payload) {
    return request('/login',{
        method: 'POST',
        headers:{"Content-Type":"application/json"},
        body:JSON.stringify(payload)
    });
}
```

src/pages/rooms/index.js

```
import React,{Component} from 'react';
import styled from 'styled-components';
import {connect} from 'dva';
import {Form,Input,message,Button,Layout,Menu,Breadcrumb,Card,Row,Col} from 'antd';
import logo from '../../assets/logo.png';
import styles from './index.less';
import Link from 'umi/link';
const {Header,Content,Footer}=Layout;
class Rooms extends Component{
    handleSubmit=(event) => {
        event.preventDefault();
        this.loginForm.props.form.validateFields((err,values) => {
            if (err) {
                return message.warn('登录信息输入不合法!请重新输入!');
            } else {
                this.props.dispatch({type:'login/login',payload:values});
            }
        });
    }
    handleKeywordChange=(event) => {
        this.props.dispatch({type: 'rooms/setKeyword',payload: event.target.value});
    }
    createRoom=() => {
        this.props.dispatch({type: 'rooms/createRoom'});
    }
    render() {
        const {list=[],user}=this.props;
        return (
            <Layout>
                <Header>
                    <img src={logo} className={styles.logo}/>
                    <Menu
                        theme="dark"
```

```
                        mode="horizontal"
                        defaultSelectedKeys={['/rooms']}
                        style={{lineHeight:'64px'}}
                    >
                        <Menu.Item key="/rooms">
                        房间列表
                                用户: {user&&user.name}
                        Menu.Item>
                    Menu>
                Header>
                <Content style={{padding:'0 50px'}}>
                    <Breadcrumb style={{margin: '16px 0'}}>
                        <Breadcrumb.Item>房间列表: {user&&user.name}Breadcrumb.Item>
                    Breadcrumb>
                    <SearchForm
                          createRoom={this.createRoom}
                          handleKeywordChange={this.handleKeywordChange}
                          size={list.length}
                        />
                    <Card>
                        <Row  gutter={16}>
                            {
                                list.map(room => (
                                    <Col key={room._id} span={6} style={{padding:'5px 0'}}>
                                        <Link to={`/messages/${room._id}`}>
                                            <Card
                                                hoverable
                                                style={{ width: 240 }}
                                                cover={<img alt="example" src="https://os.alipayobjects.com/rmsportal/QBnOOoLaAfKPirc.png" />}
                                            >
                                                <Card.Meta
                                                    title={room.name}
                                                />
                                            Card>
                                        Link>
                                    Col>
                                ))
                            }
                        Row>
                    Card>
                Content>
                <Footer style={{textAlign:'center'}}>
                    珠峰培训 @2018
                Footer>
            Layout>
        )
    }
}
class SearchForm extends Component{
    render() {
        const {handleKeywordChange,createRoom,size}=this.props;
        const formItemLayout={
            labelCol: {span: 8},
            wrapperCol:{span:16}
        }
        return (
            <Form>
                <Form.Item label="请输入房间名关键字" {...formItemLayout}>
                    <Row gutter={18}>
                        <Col span={12}><Input placeholder="请输入关键字" onChange={handleKeywordChange} />Col>
                        <Col span={12}>
                            {size==0&&<Button onClick={createRoom}>创建房间Button>}

                        Col>
                    Row>
                Form.Item>
            Form>
        )
    }
}

export default connect(
    state => ({
        list: state.rooms.list.filter(item => {
            return item.name&&item.name.indexOf(state.rooms.keyword)!=-1;
        }),
        user:state.login.user
    })
)(Rooms);
```

pages/messages/$id.js

```
import React,{Component} from 'react';
import {Layout,Row,Col,Menu,Icon,Breadcrumb,Input,List,Avatar,Spin,Button} from 'antd';
import styles from './index.less';
import messages from './models/messages';
import {connect} from 'dva';
const {Header,Sider,Footer,Content}=Layout;
class Messages extends Component{
    state={collapsed: false}
    onCollapse=() => {
        this.setState({collapsed:!this.state.collapsed});
    }
    handleKeyDown=(event) => {
        let code=event.keyCode;
        let content = event.target.value;
        if (code == 13 && content) {
            this.props.dispatch({
                type: 'messages/addMessage',
                payload:{content}
            });
        }
    }
    render() {
        let {messages,users,room,user}=this.props;
        console.log(this.props);
        return (
            <Layout style={{ minHeight: '100vh' }}>
                <Sider
                    collapsible
                    collapsed={this.state.collapsed}
                    onCollapse={this.onCollapse}
                >
                    <h3 className={styles['room-title']} >{room.name}h3>
                    <Menu style={{padding:'10px'}} theme="dark" defaultSelectedKeys={['1']} mode="inline">
                        {
                            users.map(user => (
                                <Menu.Item key={user._id}>
                                    <img src={user.avatar} style={{width:32,height:32,borderRadius:'5px'}}/>
                                    <span style={{marginLeft: 15}}>{user.name}span>
                                Menu.Item>
                            ))
                        }
                    Menu>
                Sider>
                <Layout>
                    <Content style={{margin: '16px',backgroundColor: '#FFF'}}>
                        <List
                            style={{padding:15}}
                            itemLayout="horizontal"
                            dataSource={messages}
                            renderItem={
                                message => (
                                    <List.Item actions={[]}>
                                        <List.Item.Meta
                                            avatar={<Avatar src={message.user.avatar} />}
                                            title={message.user.name}
                                            description={message.user.email}
                                        />
                                        <div>{message.content}div>
                                    List.Item>
                                )
                            }
                        >
                        List>
                    Content>
                    <Footer style={{textAlign: 'center'}}>
                        <Row>
                            <Col span={3}>
                                <img src={user && user.avatar} style={{height:30,height:30,borderRadius:5}}/>
                                {user&& user.name}说:
                            Col>
                            <Col span={21}><Input type="text" onKeyDown={this.handleKeyDown}/>Col>
                        Row>
                    Footer>
                Layout>
            Layout>
        )
    }
}
export default connect(
    state => ({
        ...state.messages,
        user:state.login.user
    })
)(Messages);
```

src/pages/messages/models/messages.js

```
import pathToRegexp from 'path-to-regexp';
import io from 'socket.io-client';
import {routerRedux} from 'dva/router';
let client;
export default {

    namespace: 'messages',

    state: {
        room: null,
        users:[],
        messages:[]
    },

    subscriptions: {
      setup({dispatch,history}) {
          history.listen(({pathname,query}) => {
              let result=pathToRegexp('/messages/:id').exec(pathname);
              if (result) {
                  let room=result[1];
                  dispatch({type: 'setRoom',payload: room});
                  dispatch({type: 'login/validate'});
                  let socket=io('http://127.0.0.1:7001',{
                      query: {token:localStorage.getItem('token')||'',room}
                  });
                  client=socket;
                  socket.on('connect',() => {
                      socket.emit('getRoom',room);
                  });
                  socket.on('room',(room) => {
                      dispatch({type:'room',payload:room});
                  });
                  socket.on('messageAdded',message => {
                      dispatch({type:'messageAdded',payload:message});
                  });
                    socket.on('online',user => {
                      dispatch({type:'addUser',payload:user});
                  });
                  socket.on('offline',id => {
                      dispatch({type:'delUser',payload:id});
                  });
                  socket.on('needLogin',() => {
                      dispatch(routerRedux.push('/login'));
                    });
                    socket.on('disconnect',() => {
                        dispatch({type: 'login/logout'});
                    });
                    socket.on('error',() => {
                      dispatch(routerRedux.push('/login'));
                    });
              }
          });
      }
    },

    effects: {
      *addMessage({ payload }, { call, put,select }) {
          const {login:{user},messages:{room}}=yield select(state => state);
          let message=payload;
          message.user=user._id;
          message.room=room;
          client.emit('addMessage',message);
      }
    },

    reducers: {
      setRoom(state, action) {
          return { ...state, room:action.payload};
      },
      room(state,{payload}) {
          return { ...state, ...payload};
      },
      messageAdded(state,action) {
          return {...state,messages:[...state.messages,action.payload]};
      },
        addUser(state,{payload}) {
            let existUser=state.users.find(user => user._id==payload._id);
          return existUser?state:{...state,users:[...state.users,payload]};
      },
        delUser(state,{payload}) {
          return {...state,users:state.users.filter(item=>item._id!= payload)};
      }
    }
};
```

6. 支持表情

src/pages/messages/$id.js

```
import React,{Component} from 'react';
import {Layout,Row,Col,Menu,Icon,Breadcrumb,Input,List,Avatar,Spin,Button,Popover,Card} from 'antd';
import styles from './index.less';
import messages from './models/messages';
import {connect} from 'dva';
import expressions from './expressions';
import styled from 'styled-components';
import face from '../../assets/face.png';
const {Header,Sider,Footer,Content}=Layout;
class Messages extends Component{
    state={collapsed: false,inputVal:''}
    onCollapse=() => {
        this.setState({collapsed:!this.state.collapsed});
```

```
    }
    sendMessage=(event) => {
        let content = event.target.value;
        this.props.dispatch({
            type: 'messages/addMessage',
            payload:{content}
        });
    }
    handleInputChange=(event) => {
        this.setState({
            inputVal:event.target.value
        });
    }
    fillFace=(event) => {
        const {index}=event.currentTarget.dataset;
        this.setState({
            inputVal:`${this.state.inputVal}#(${index})`
        });
    }
    render() {
        let {messages,users,room,user}=this.props;
        let faces=(
            <Row style={{width:512}}>
                {
                    expressions.default.map((item,index) => (
                        <Col key={index} span={3}>
                            <div
                                data-index={index}
                                onClick={this.fillFace}
                                style={{width: '64px',height: '64px',backgroundImage: `url(${face})`,backgroundPosition: `left ${-64*index+'px'}`}}>div>
                        Col>
                    ))
                }
            Row>
        )
        let facePanel=(
            <Popover content={faces} title="表情" trigger="click">
                表情
            Popover>
        )
        return (
            <Layout style={{ minHeight: '100vh' }}>
                <Sider
                    collapsible
                    collapsed={this.state.collapsed}
                    onCollapse={this.onCollapse}
                >
                    <h3 className={styles['room-title']} >{room.name}h3>
                    <Menu style={{padding:'10px'}} theme="dark" defaultSelectedKeys={['1']} mode="inline">
                        {
                            users.map(user => (
                                <Menu.Item key={user._id}>
                                    <img src={user.avatar} style={{width:32,height:32,borderRadius:'5px'}}/>
                                    <span style={{marginLeft: 15}}>{user.name}span>
                                Menu.Item>
                            ))
                        }
                    Menu>
                Sider>
                <Layout>
                    <Content style={{margin: '16px',backgroundColor: '#FFF'}}>
                        <List
                            style={{padding:15}}
                            itemLayout="horizontal"
                            dataSource={messages}
                            renderItem={
                                message => {
                                    let content=message.content;
                                    if (!content || content =='undefined') return  <div>div>;
                                    content=content.replace(/#\((\d+)\)/g,function (matched,index) {
                                        let offset=-64*parseInt(index)+'px';
                                        return `<span
                                                  style="display:inline-block;width:64px;height:64px;background-image:url(${face});background-position:left
${offset}"
                                                >span>`
                                    });
                                    return (
                                        <List.Item actions={[]}>
                                        <List.Item.Meta
                                            avatar={<Avatar src={message.user.avatar} />}
                                            title={message.user.name}
                                            description={message.user.email}
                                            />
                                            <div dangerouslySetInnerHTML={{__html:content}}>div>
                                    List.Item>
                                    )
                                }
                            }
                        >
                        List>
                    Content>
                    <Footer style={{textAlign: 'center'}}>
                        <Row>
                            <Col span={3}>
                                <img src={user && user.avatar} style={{height:30,height:30,borderRadius:5}}/>
                                {user&& user.name}说:
                            Col>
                            <Col span={19}>
                                <Input
                                    onChange={this.handleInputChange}
                                    value={this.state.inputVal}
                                    addonBefore={facePanel}
                                    placeholder="请输入信息"
                                    onPressEnter={this.sendMessage}
```

```
                            />
                        Col>
                    Row>
                Footer>
            Layout>
        Layout>
    )
  }
}

export default connect(
  state => ({
    ...state.messages,
    user:state.login.user
  })
)(Messages);
```

src/pages/messages/expressions.js

```
export default {
  default: [
    '呵呵', '哈哈', '吐舌', '啊', '酷', '怒', '开心', '汗', '泪', '黑线',
    '鄙视', '不高兴', '真棒', '钱', '疑问', '阴险', '吐', '咦', '委屈', '花心',
    '呼', '笑眼', '冷', '太开心', '滑稽', '勉强', '狂汗', '乖', '睡觉', '惊哭',
    '升起', '惊讶', '喷', '爱心', '心碎', '玫瑰', '礼物', '星星月亮', '太阳', '音乐',
    '灯泡', '蛋糕', '彩虹', '钱币', '咖啡', 'haha', '胜利', '大拇指', '弱', 'ok',
  ],
};
```

参考