link: null
title: 珠峰架构师成长计划
description: tsconfig.json
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=132 sentences=297, words=3030

1.课程大纲

- UMI4应用开发实战
- UMI4+AntDesginProV6应用开发实战
- 从零实现UMI4插件系统和核心插件

2.项目初始化

```
mkdir umi4project
cd umi4project
pnpm init -y
pnpm config get registry
pnpm config set registry https://registry.npmmirror.com/
pnpm i @ant-design/icons @ant-design/pro-components @umijs/max ahooks antd@4  jsonwebtoken
```

```
max generate
```

3.约定式路由

```
mdir src
max g
```

```
max g
```

```
{
  "scripts": {
    "dev": "max dev",
    "build": "max build"
  },
}
```

```
/node_modules
/src/.umi
/dist
```

tsconfig.json

```
{
  "extends": "./src/.umi/tsconfig.json",
+ "compilerOptions": {
+   "noImplicitAny": false
+ }
}
```

```
pnpm dev
```

4.配置式路由

config\config.ts

```
import { defineConfig } from '@umijs/max';
+import routes from "./routes";
export default defineConfig({
  npmClient: 'pnpm',
+ routes
});
```

config\routes.ts

```
export default [
  { path: '/', redirect: '/home' },
  { icon: 'HomeOutlined', name: '首页', path: '/home', component: './home/index' },
  { icon: 'ProfileOutlined', name: '个人中心', path: '/profile', component: './profile/index' },
]
```

src\pages\home\index.tsx

```
import styles from './index.less';
+import { Link, history, useNavigate } from '@umijs/max';
+import { Button } from 'antd';
export default function Page() {
  let navigate = useNavigate();
  return (

+      首页
+      个人中心
+       history.push('/profile')}>个人中心
+       navigate('/profile')}>个人中心

  );
}
```

5.tailwindcss

```
max g
```

src\pages\home\index.tsx

```
import { Link, history, useNavigate } from '@umijs/max';
import { Button } from 'antd';
export default function Page() {
  let navigate = useNavigate();
  return (

+      首页
      个人中心
       history.push('/profile')}>个人中心
       navigate('/profile')}>个人中心

  );
}
```

src\pages\profile\index.tsx

```
export default function Page() {
  return (

+      个人中心

  );
}
```

6.支持布局

```
import { defineConfig } from "@umijs/max";
import routes from "./routes";
export default defineConfig({
  npmClient: "pnpm",
  routes,
  tailwindcss: {},
+ layout: {
+   title: "UMI",
+   locale: false
+ },
+ antd: {},
});
```

7.支持子路由

config\routes.ts

```
export default [
  { path: '/', redirect: '/home' },
  { icon: 'HomeOutlined', name: '首页', path: '/home', component: './home/index' },
  { icon: 'ProfileOutlined', name: '个人中心', path: '/profile', component: './profile/index' },
+ {
+   icon: 'UserOutlined',
+   name: '用户管理',
+   path: '/user',
+   component: './user/index',
+   routes: [
+     { name: '添加用户', path: '/user/add', component: './user/add/index' },
+     { name: '用户列表', path: '/user/list', component: './user/list/index' },
+     { name: '用户详情', path: '/user/detail/:id', component: './user/detail/index', hideInMenu: true },
+   ],
+ },
]
```

src\pages\user\index.tsx

```
import { PageContainer } from '@ant-design/pro-components';
import { Outlet } from '@umijs/max';
export default function () {
    return (
        <PageContainer>
            <Outlet />
        PageContainer>
    );
}
```

src\pages\user\list\index.tsx

```
import { List } from 'antd';
import { Link } from '@umijs/max';

export default function () {
    const data = {
        list: [
            { id: 1, username: 'root' },
            { id: 2, username: 'admin' },
            { id: 3, username: 'member' }
        ]
    }
    return (
        <List
            header={<div>用户列表div>}
            footer={<div>共计{data?.list?.length}div>}
            bordered
            dataSource={data?.list}
            renderItem={(user: any) => (
                <List.Item>
                    <Link to={`/user/detail/${user.id}`} state={user}> {user.username}Link>
                List.Item>
            )}
        />
    );
}
```

src\pages\user\add\index.tsx

```
import { Form, Input, Button, Row, Col } from 'antd';
export default function () {
    const onFinish = (values: any) => {
        console.log(values);
    };
    return (
        <Row >
            <Col offset={8} span={8}>
                <Form
                    labelCol={{ span: 4 }}
                    wrapperCol={{ span: 20 }}
                    onFinish={onFinish}
                >
                    <Form.Item
                        label="用户名"
                        name="username"
                        rules={[{ required: true, message: '请输入用户名' }]}
                    >
                        <Input />
                    Form.Item>
                    <Form.Item
                        label="密码"
                        name="password"
                        rules={[{ required: true, message: '请输入密码' }]}
                    >
                        <Input.Password />
                    Form.Item>
                    <Form.Item
                        label="手机号"
                        name="phone"
                        rules={[{ required: true, message: '请输入手机号' }]}
                    >
                        <Input />
                    Form.Item>
                    <Form.Item wrapperCol={{ offset: 8, span: 16 }}>
                        <Button type="primary" htmlType="submit" >
                            提交
                        Button>
                    Form.Item>
                Form>
            Col>
        Row>

    );
}
```

src\pages\user\detail\index.tsx

```
import { Descriptions } from 'antd';
import { useLocation } from '@umijs/max';
export default function (props: any) {
    const location = useLocation();
    let user = location.state as API.User;
    return (
        <Descriptions title="用户信息">
            <Descriptions.Item label="用户名">{user?.username}Descriptions.Item>
            <Descriptions.Item label="手机号">{user?.phone}Descriptions.Item>
        Descriptions>
    );
}
```

8.请求用户列表

```
import { defineConfig } from "@umijs/max";
import routes from "./routes";
export default defineConfig({
  npmClient: "pnpm",
  routes,
  tailwindcss: {},
  layout: {
    title: "UMI",
    locale: false
  },
  antd: {},
+ request: {},
+ model: {},
+ initialState: {},
+ proxy: {
+   '/api/': {
+     target: 'http://127.0.0.1:7001/',
+     changeOrigin: true
+   }
+ }
});
```

src\pages\user\list\index.tsx

```
import { List } from 'antd';
import { Link, useModel } from '@umijs/max';
export default function () {
+   const { data, loading } = useModel('user.model');
    return (
        用户列表}
            footer={共计{data?.list?.length}}
            bordered
            dataSource={data?.list}
            renderItem={(user: any) => (

                         {user.username}

            )}
        />
    );
}
```

src\pages\user\model.ts

```
import { useRequest } from 'ahooks';
import { getUser } from '@/services/user';

export default () => {
  const { data, loading, refresh } = useRequest(getUser);
  return {
    data,
    refresh,
    loading
  };
};
```

src\services\typings.d.ts

```
declare namespace API {
    export interface ListResponse {
        data?: ListData;
        errorCode: string;
        errorMessage: string;
        errorType: number;
        success?: boolean;
    }

    export interface ListData {
        list?: T[];
        total?: number;
    }

    export interface User {
        id?: number;
        password?: string;
        phone?: string;
        username?: string;
        role?: string;
    }
    export interface SigninUser {
        username?: string;
        password?: string;
    }
    export interface SignupUser {
        password?: string;
        phone?: string;
        username?: string;
        role_id?: number;
    }
}
```

src\services\user.ts

```
import { request } from '@umijs/max';
export async function getUser() {
  return request>('/api/user', {
    method: 'GET'
  });
}
```

src\app.ts

```
import { notification, message } from 'antd';

enum ErrorShowType {
    SILENT = 0,
    WARN_MESSAGE = 1,
    ERROR_MESSAGE = 2,
    NOTIFICATION = 3
}

const errorHandler = (error: any) => {
    if (error.name === 'BizError') {
        const errorInfo = error.info;
        if (errorInfo) {
            const { errorMessage, errorCode, showType } = errorInfo;
            switch (errorInfo.showType) {
                case ErrorShowType.SILENT:
                    break;
                case ErrorShowType.WARN_MESSAGE:
                    message.warn(errorMessage);
                    break;
                case ErrorShowType.ERROR_MESSAGE:
                    message.error(errorMessage);
                    break;
                case ErrorShowType.NOTIFICATION:
                    notification.open({
                        description: errorMessage,
                        message: errorCode,
                    });
                    break;
                default:
                    message.error(errorMessage);
            }
        }
    } else if (error.response) {
        message.error(`响应状态码:${error.response.status}`);
    } else if (error.request) {
        message.error('没有收到响应');
    } else {
        message.error('请求发送失败');
    }
};
const errorThrower = (res: any) => {
    const { success, data, errorCode, errorMessage } = res;
    if (!success) {
        const error: any = new Error(errorMessage);
        error.name = 'BizError';
        error.info = { errorCode, errorMessage, data };
        throw error;
    }
}
export const request = {
    timeout: 3000,
    headers: {
        ['Content-Type']: 'application/json',
        ['Accept']: 'application/json',
        credentials: 'include'
    },
    errorConfig: {
        errorThrower,
        errorHandler
    },
    requestInterceptors: [(url, options) => {
        return { url, options }
    }],
    responseInterceptors: [(response, options) => {
        return response.data;
    }]
};

export const layout = () => {
    return {
        title: 'UMI4',
        logo: 'https://img.alicdn.com/tfs/TB1YHEpwUT1gK0jSZFhXXaAtVXa-28-27.svg',
    };
};
```

9.添加用户

src\pages\user\add\index.tsx

```
import { Form, Input, Button, Row, Col } from 'antd';
+import { useRequest } from 'ahooks';
+import { addUser } from '@/services/user';
+import { useNavigate, useModel } from '@umijs/max';
+import { useEffect } from 'react';

export default function () {
+   const navigate = useNavigate();
+   const { refresh } = useModel('user.model');
+   const { data, loading, run } = useRequest(addUser, {
+       manual: true,
+       onSuccess: refresh
+   });
    const onFinish = (values: any) => {
+       run(values);
    };
+   useEffect(() => {
+       if (data) {
+           navigate('/user/list');
+       }
+   }, [data]);
    return (

                            提交

    );
}
```

src\services\user.ts

```
import { request } from '@umijs/max';

export async function getUser() {
  return request>('/api/user', {
    method: 'GET'
  });
}
+export async function addUser(
+  body?: API.User
+) {
+  return request('/api/user', {
+    method: 'POST',
+    headers: {
+      'Content-Type': 'application/json',
+    },
+    data: body
+  });
+}
```

10.用户注册

config\routes.ts

```
export default [
  { path: '/', redirect: '/home' },
  { icon: 'HomeOutlined', name: '首页', path: '/home', component: './home/index' },
  { icon: 'ProfileOutlined', name: '个人中心', path: '/profile', component: './profile/index' },
  {
    icon: 'UserOutlined',
    name: '用户管理',
    path: '/user',
    component: './user/index',
    routes: [
      { name: '添加用户', path: '/user/add', component: './user/add/index' },
      { name: '用户列表', path: '/user/list', component: './user/list/index' },
      { name: '用户详情', path: '/user/detail/:id', component: './user/detail/index', hideInMenu: true },
    ],
  },
+ {
+   name: '注册',
+   path: '/signup',
+   component: './signup/index',
+   hideInMenu: true,
+   layout: false
+ }
]
```

src\app.ts

```
import { notification, message } from 'antd';
+import { history } from '@umijs/max';

enum ErrorShowType {
    SILENT = 0,
    WARN_MESSAGE = 1,
    ERROR_MESSAGE = 2,
    NOTIFICATION = 3
}

const errorHandler = (error: any) => {
    if (error.name
        const errorInfo = error.info;
        if (errorInfo) {
            const { errorMessage, errorCode, showType } = errorInfo;
            switch (errorInfo.showType) {
                case ErrorShowType.SILENT:
                    break;
                case ErrorShowType.WARN_MESSAGE:
                    message.warn(errorMessage);
                    break;
                case ErrorShowType.ERROR_MESSAGE:
                    message.error(errorMessage);
                    break;
                case ErrorShowType.NOTIFICATION:
                    notification.open({
                        description: errorMessage,
                        message: errorCode,
                    });
                    break;
                default:
                    message.error(errorMessage);
            }
        }
    } else if (error.response) {
        message.error(`响应状态码:${error.response.status}`);
    } else if (error.request) {
        message.error('没有收到响应');
    } else {
        message.error('请求发送失败');
    }
};
const errorThrower = (res: any) => {
    const { success, data, errorCode, errorMessage } = res;
    if (!success) {
        const error: any = new Error(errorMessage);
        error.name = 'BizError';
        error.info = { errorCode, errorMessage, data };
        throw error;
    }
}
export const request = {
    timeout: 3000,
    headers: {
        ['Content-Type']: 'application/json',
        ['Accept']: 'application/json',
        credentials: 'include'
    },
    errorConfig: {
        errorThrower,
        errorHandler
    },
    requestInterceptors: [(url, options) => {
        return { url, options }
    }],
    responseInterceptors: [(response, options) => {
        return response.data;
    }]
};
+interface InitialState {
+    currentUser: API.User | null
+}
+
+export async function getInitialState() {
+    let initialState: InitialState = {
+        currentUser: null
+    }
+    return initialState;
+}
export const layout = ({ initialState }) => {
    return {
        title: 'UMI4',
        logo: 'https://img.alicdn.com/tfs/TB1YHEpwUT1gK0jSZFhXXaAtVXa-28-27.svg',
+       onPageChange(location) {
+           const { currentUser } = initialState;
+           if (!currentUser && location.pathname !== '/signin') {
+               history.push('/signup');
+           }
+       }
    };
};
```

src\constants\index.ts

```
enum RoleCodes {
    root = 'root',
    admin = 'admin',
    member = 'member'
}
const ROLES = [
    {
        code: RoleCodes.root,
        name: '超级管理员'
    },
    {
        code: RoleCodes.admin,
        name: '管理员'
    },
    {
        code: RoleCodes.member,
        name: '普通成员'
    }
]

export {
    RoleCodes,
    ROLES
}
```

src\pages\signup\index.tsx

```
import { Form, Input, Button, Card, Row, Col, Spin, Select } from 'antd';
import { useRequest } from 'ahooks';
import { signup } from '@/services/auth';
import { history, Link } from '@umijs/max';
import { ROLES } from '@/constants';

export default function () {
    const { loading, run } = useRequest(signup, {
        manual: true,
        onSuccess() {
            history.push('/signin')
        }
    });
    const onFinish = (values: any) => {
        run(values);
    };
    const onFinishFailed = (errorInfo: any) => {
        console.log('Failed:', errorInfo);
    };
    return (
        <Row className='h-screen bg-gray-200' align='middle'>
            <Col offset={8} span={8} >
                <Card title="请登录" extra={<Link to="/signin">去登录Link>}>
                    <Spin spinning={loading}>
                        <Form
                            labelCol={{ span: 4 }}
                            wrapperCol={{ span: 20 }}
                            onFinish={onFinish}
                            onFinishFailed={onFinishFailed}
                        >
                            <Form.Item
                                label="用户名"
                                name="username"
                                rules={[{ required: true, message: '请输入用户名' }]}
                            >
                                <Input />
                            Form.Item>
                            <Form.Item
                                label="密码"
                                name="password"
                                rules={[{ required: true, message: '请输入密码' }]}
                            >
                                <Input.Password />
                            Form.Item>
                            <Form.Item
                                label="手机号"
                                name="phone"
                                rules={[{ required: true, message: '请输入手机号' }]}
                            >
                                <Input />
                            Form.Item>
                            <Form.Item
                                label="角色"
                                name="role"
                                rules={[{ required: true, message: '请选择角色' }]}
                            >
                                <Select
                                >
                                    {
                                        ROLES.map(role => {
                                            return (
                                                <Select.Option value={role.code} key={role.code}>
                                                    {role.name}
                                                Select.Option>
                                            )
                                        })
                                    }
                                Select>
                            Form.Item>
                            <Form.Item wrapperCol={{ offset: 8, span: 16 }}>
                                <Button type="primary" htmlType="submit" >
                                    提交
                                Button>
                            Form.Item>
                        Form>
                    Spin>
                Card>
            Col>
        Row>
    );
}
```

src\services\auth.ts

```
import { request } from '@umijs/max';

export async function signup(
  body?: API.SignupUser
) {
  return request('/api/user', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    data: body
  });
}
```

11.用户登录

config\routes.ts

```
export default [
  { path: '/', redirect: '/home' },
  { icon: 'HomeOutlined', name: '首页', path: '/home', component: './home/index' },
  { icon: 'ProfileOutlined', name: '个人中心', path: '/profile', component: './profile/index' },
  {
    icon: 'UserOutlined',
    name: '用户管理',
    path: '/user',
    component: './user/index',
    routes: [
      { name: '添加用户', path: '/user/add', component: './user/add/index' },
      { name: '用户列表', path: '/user/list', component: './user/list/index' },
      { name: '用户详情', path: '/user/detail/:id', component: './user/detail/index', hideInMenu: true },
    ],
  },
  {
    name: '注册',
    path: '/signup',
    component: './signup/index',
    hideInMenu: true,
    layout: false
  },
+ {
+   name: '登录',
+   path: '/signin',
+   component: './signin/index',
+   hideInMenu: true,
+   layout: false
+ }
]
```

src\app.ts

```
import { notification, message } from 'antd';
import { history } from '@umijs/max';
+import { decode } from 'jsonwebtoken';

enum ErrorShowType {
    SILENT = 0,
    WARN_MESSAGE = 1,
    ERROR_MESSAGE = 2,
    NOTIFICATION = 3
}

const errorHandler = (error: any) => {
    if (error.name
        const errorInfo = error.info;
        if (errorInfo) {
            const { errorMessage, errorCode, showType } = errorInfo;
            switch (errorInfo.showType) {
                case ErrorShowType.SILENT:
                    break;
                case ErrorShowType.WARN_MESSAGE:
                    message.warn(errorMessage);
                    break;
                case ErrorShowType.ERROR_MESSAGE:
                    message.error(errorMessage);
                    break;
                case ErrorShowType.NOTIFICATION:
                    notification.open({
                        description: errorMessage,
                        message: errorCode,
                    });
                    break;
                default:
                    message.error(errorMessage);
            }
        }
    } else if (error.response) {
        message.error(`响应状态码:${error.response.status}`);
    } else if (error.request) {
        message.error('没有收到响应');
    } else {
        message.error('请求发送失败');
    }
};
const errorThrower = (res: any) => {
    const { success, data, errorCode, errorMessage } = res;
    if (!success) {
        const error: any = new Error(errorMessage);
        error.name = 'BizError';
        error.info = { errorCode, errorMessage, data };
        throw error;
    }
}
export const request = {
    timeout: 3000,
    headers: {
        ['Content-Type']: 'application/json',
        ['Accept']: 'application/json',
        credentials: 'include'
    },
    errorConfig: {
        errorThrower,
        errorHandler
    },
    requestInterceptors: [(url, options) => {
+        const token = localStorage.getItem('token');
+        if (token) {
+            options.headers.authorization = token
+        }
+        return { url, options }
+    }],
    responseInterceptors: [(response, options) => {
        return response.data;
    }]
};
interface InitialState {
    currentUser: API.User | null
}

export async function getInitialState() {
    let initialState: InitialState = {
        currentUser: null
    }
+   const token = localStorage.getItem('token');
+   if (token) {
+       initialState.currentUser = decode(token)
+   }
    return initialState;
}
export const layout = ({ initialState }) => {
    return {
        title: 'UMI4',
        logo: 'https://img.alicdn.com/tfs/TB1YHEpwUT1gK0jSZFhXXaAtVXa-28-27.svg',
        onPageChange(location) {
            const { currentUser } = initialState;
+           if (!currentUser && location.pathname !== '/signin') {
+               history.push('/signin');
+           }
        }
    };
};
```

src\pages\signin\index.tsx

```
import { Form, Input, Button, Card, Row, Col, Spin } from 'antd';
import { useRequest } from 'ahooks';
import { signin } from '@/services/auth';
import { useModel, history, Link } from '@umijs/max';
import { useEffect } from 'react';
import { decode } from 'jsonwebtoken';

export default function () {
    const { initialState, setInitialState } = useModel('@@initialState');
    const { loading, run } = useRequest(signin, {
        manual: true,
        onSuccess(result) {
            localStorage.setItem('token', result);
            const currentUser = decode(result);
            setInitialState({ currentUser });
        }
    });
    useEffect(() => {
        if (initialState?.currentUser)
            history.push('/')
    }, [initialState]);
    const onFinish = (values: any) => {
        run(values);
    };
    const onFinishFailed = (errorInfo: any) => {
        console.log('Failed:', errorInfo);
    };
    return (
        <Row className='h-screen bg-gray-200' align='middle'>
            <Col offset={8} span={8} >
                <Card title="请登录" extra={<Link to="/signup">去注册Link>}>
                    <Spin spinning={loading}>
                        <Form
                            labelCol={{ span: 4 }}
                            wrapperCol={{ span: 20 }}
                            onFinish={onFinish}
                            onFinishFailed={onFinishFailed}
                        >
                            <Form.Item
                                label="用户名"
                                name="username"
                                rules={[{ required: true, message: '请输入用户名' }]}
                            >
                                <Input />
                            Form.Item>
                            <Form.Item
                                label="密码"
                                name="password"
                                rules={[{ required: true, message: '请输入密码' }]}
                            >
                                <Input.Password />
                            Form.Item>
                            <Form.Item wrapperCol={{ offset: 8, span: 16 }}>
                                <Button type="primary" htmlType="submit" >
                                    提交
                                Button>
                            Form.Item>
                        Form>
                    Spin>
                Card>
            Col>
        Row>
    );
}
```

src\services\auth.ts

```
import { request } from '@umijs/max';

export async function signup(
  body?: API.SignupUser
) {
  return request('/api/user', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    data: body
  });
}

+export async function signin(
+  body?: API.SigninUser
+) {
+  return request('/api/signin', {
+    method: 'POST',
+    headers: {
+      'Content-Type': 'application/json',
+    },
+    data: body
+  });
+}
```

12.用户退出

src\pages\profile\index.tsx

```
+import { PageContainer } from '@ant-design/pro-components';
+import HeaderMenu from '@/components/HeaderMenu';

export default function Page() {
  return (
+
+
+
  );
}
```

src\components\HeaderMenu\index.tsx

```
import { Dropdown, Menu } from 'antd'
import { LoginOutlined } from '@ant-design/icons';
import { useModel, history } from '@umijs/max';

export default function HeaderMenu() {
  const { initialState, setInitialState } = useModel('@@initialState');
  const logoutClick = () => {
    localStorage.removeItem('token');
    setInitialState({ currentUser: null });
    history.push('/signin');
  }
  const menu = (
    <Menu items={[
      {
        key: 'logout',
        label: (
          <span>退出span>
        ),
        icon: <LoginOutlined />,
        onClick: logoutClick
      }
    ]} />
  )
  return (
    <Dropdown.Button overlay={menu}>
      {initialState?.currentUser?.username}
    Dropdown.Button>
  )
}
```

13.路由权限

config\config.ts

```
import { defineConfig } from "@umijs/max";
import routes from "./routes";
export default defineConfig({
  npmClient: "pnpm",
  routes,
  tailwindcss: {},
  layout: {
    title: "UMI",
    locale: false
  },
  antd: {},
  request: {},
  model: {},
  initialState: {},
  proxy: {
    '/api/': {
      target: 'http://127.0.0.1:7001/',
      changeOrigin: true
    }
  },
+ access: {}
});
```

config\routes.ts

```
export default [
  { path: '/', redirect: '/home' },
  { icon: 'HomeOutlined', name: '首页', path: '/home', component: './home/index' },
  { icon: 'ProfileOutlined', name: '个人中心', path: '/profile', component: './profile/index' },
  {
    icon: 'UserOutlined',
    name: '用户管理',
    path: '/user',
    component: './user/index',
    routes: [
+      { name: '添加用户', path: '/user/add', component: './user/add/index', access: 'adminCan' },
+      { name: '用户列表', path: '/user/list', component: './user/list/index', access: 'memberCan' },
+      { name: '用户详情', path: '/user/detail/:id', component: './user/detail/index', hideInMenu: true, access: 'memberCan' },
    ],
  },
  {
    name: '注册',
    path: '/signup',
    component: './signup/index',
    hideInMenu: true,
    layout: false
  },
  {
    name: '登录',
    path: '/signin',
    component: './signin/index',
    hideInMenu: true,
    layout: false
  },
]
```

src\access.ts

```
import { RoleCodes } from '@/constants';
export default function (initialState) {
    const role = initialState?.currentUser?.role;
    return {
        rootCan: role === RoleCodes.root,
        adminCan: [RoleCodes.root, RoleCodes.admin].includes(role),
        memberCan: [RoleCodes.root, RoleCodes.admin, RoleCodes.member].includes(role),
    };
}
```

14.按钮权限

src\pages\user\list\index.tsx

```
+import { List, Button } from 'antd';
+import { Link, useModel, useAccess } from '@umijs/max';
+import { useRequest } from 'ahooks';
+import { deleteUser } from '@/services/user';

export default function () {
    const { data, loading, refresh } = useModel('user.model');
+   const { run } = useRequest(deleteUser, {
+       manual: true,
+       onSuccess: refresh
+   });
+   const access = useAccess();
    return (
        用户列表}
            footer={共计{data?.list?.length}}
            bordered
            dataSource={data?.list}
            renderItem={(user: any) => (

                     {user.username}
+
+                       onClick={() => run(user.id)}
+                       size='small'
+                       type="primary"
+                       disabled={!access.adminCan}
+                       loading={loading}>
+                       删除
+

            )}
        />
    );
}
```

src\services\user.ts

```
import { request } from '@umijs/max';

export async function getUser() {
  return request>('/api/user', {
    method: 'GET'
  });
}
export async function addUser(
  body?: API.User
) {
  return request('/api/user', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    data: body
  });
}
+export async function deleteUser(id) {
+  return request(`/api/user/${id}`, {
+    method: 'DELETE'
+  });
+}
```