react全家桶学习笔记1：[学习笔记1-路由基础配置](file:///G:\C以及其他教程\Web开发2023\web开发2023视频教程\凡云1\react零基础学习资料+学习笔记\react全家桶笔记\react全家桶学习笔记1-路由基础配置.docx)

react全家桶学习笔记2：[学习笔记2-首页布局和菜单渲染](file:///G:\C以及其他教程\Web开发2023\web开发2023视频教程\凡云1\react零基础学习资料+学习笔记\react全家桶笔记\react全家桶学习笔记2-首页布局和菜单渲染.docx)

react全家桶学习笔记3：[学习笔记3二级路由配置](file:///G:\C以及其他教程\Web开发2023\web开发2023视频教程\凡云1\react零基础学习资料+学习笔记\react全家桶笔记\react全家桶学习笔记3-二级路由配置.docx)

react全家桶学习笔记4：学习笔记4[配置菜单默认项](file:///G:\C以及其他教程\Web开发2023\web开发2023视频教程\凡云1\react零基础学习资料+学习笔记\react全家桶笔记\react全家桶学习笔记4-设置菜单默认项.docx)

react全家桶学习笔记5：[学习笔记5 axios配置网络请求](react全家桶学习笔记5-axios设置网络请求.docx)

react全家桶学习笔记6：学习笔记6用户列表渲染

# 还是React-fam-bucketdemo项目，我们现在来做用户列表的渲染

## 1）到ant design给官网找一个自己喜欢的表格组件，把代码拷贝过来，注意一点一点的拷贝

### Users Page.jsx

|  |
| --- |
| **import *React***, {*useEffect*} **from 'react'**; **import *axios* from 'axios' import *request* from "../../../utils/request"**; **import** {*getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**; **const** columns = [  {  **title**: **'Name'**,  **dataIndex**: **'name'**,  **key**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'Age'**,  **dataIndex**: **'age'**,  **key**: **'age'**,  },  {  **title**: **'Address'**,  **dataIndex**: **'address'**,  **key**: **'address'**,  },  {  **title**: **'Tags'**,  **key**: **'tags'**,  **dataIndex**: **'tags'**,  render: (\_, { tags }) => (  <>  {tags.map((tag) => {  **let** color = tag.**length** > 5 ? **'geekblue'** : **'green'**;  **if** (tag === **'loser'**) {  color = **'volcano'**;  }  **return** (  <**Tag color=**{color} **key=**{tag}>  {tag.toUpperCase()}  </**Tag**>  );  })}  </>  ),  },  {  **title**: **'Action'**,  **key**: **'action'**,  render: (\_, record) => (  <**Space size="middle"**>  <**a**>Invite {record.**name**}</**a**>  <**a**>Delete</**a**>  </**Space**>  ),  }, ]; **const** data = [  {  **key**: **'1'**,  **name**: **'John Brown'**,  **age**: 32,  **address**: **'New York No. 1 Lake Park'**,  **tags**: [**'nice'**, **'developer'**],  },  {  **key**: **'2'**,  **name**: **'Jim Green'**,  **age**: 42,  **address**: **'London No. 1 Lake Park'**,  **tags**: [**'loser'**],  },  {  **key**: **'3'**,  **name**: **'Joe Black'**,  **age**: 32,  **address**: **'Sydney No. 1 Lake Park'**,  **tags**: [**'cool'**, **'teacher'**],  }, ];  **const** *UsersPage* = () => {  *useEffect*(()=>{  getUsers()  },[])  **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  }   }  **return** (  <>  <**Table columns=**{columns} **dataSource=**{data} />;  </>  ); };  **export default** *UsersPage*; |

### 效果如下

|  |
| --- |
|  |

## 2）把上面表格的数据替换成我们的数据

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**; **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => (  <**Space size="middle"**>  <**a**>修改</**a**>  <**a**>删除</**a**>  </**Space**>  ),  }, ];  **const** *UsersPage* = () => {  **let** [usersData,setUsersData] = *useState*([])  *useEffect*(()=>{  getUsers()  },[])  **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData(res)  }   }  **return** (  <>  <**Table columns=**{columns} **dataSource=**{usersData} />;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |

#### 他会自动分页，不过这里由一个报错，说每一个遍历出来的孩子需要由key属性

## 3）在渲染表格的地方添加一个rowKey属性，然后错误就没有了

|  |
| --- |
|  |

## 4）我们想对角色一栏的数据处理一下，把超级管理员的文本样式突出一下，可以使用ant design的Tag组件来实现，我们到ant design里面找一个喜欢的Tag

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**; **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => (  <**Space size="middle"**>  <**a**>修改</**a**>  <**a**>删除</**a**>  </**Space**>  ),  }, ];  **const** *UsersPage* = () => {  **let** [usersData,setUsersData] = *useState*([])  *useEffect*(()=>{  getUsers()  },[])  **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData(res)  }   }  **return** (  <>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="\_id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |

#### 注意，如果你需要很多颜色，可以把箭头函数改为普通函数添加switch或者多个if分支，不过此时需要用return语句来返回

## 5）实现删除功能，需要发送请求到服务器例如我们这里是json-server，需要在users.js里面封装api，我们打开users.js,添加删除api，注意我们的路径和老师的不一样，因为我没有真正的后端，老师的是这样子的

|  |
| --- |
|  |
|  |
|  |
|  |

## 老师的后端删除是用get方法传递参数

|  |
| --- |
|  |
|  |

## 我的接口是这样子的（users.js）

|  |
| --- |
| **import *request* from "../utils/request"**;  **export let** *getUserApi* = () => ***request***.get(**"/users"**) **export let** *delUserApi* = (id) => ***request***.delete(**`/users/**${id}**`**) |

### 这里由一个坑，json-server不支持\_id,我们需要把data.json的所有\_id改为id

|  |
| --- |
| {  **"users"**: [  {  **"id"**: **"user000001"**,  **"account"**: **"xiaowang"**,  **"createDate"**: **null**,  **"email"**: **"xiaowang123@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000002"**,  **"account"**: **"xiaozhang"**,  **"createDate"**: **null**,  **"email"**: **"xiaozhang123@hotmail.com"**,  **"role"**: {  **"id"**: **"1112"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://thumbs.dreamstime.com/b/delighted-nice-women-talking-meeting-friends-delighted-nice-positive-women-looking-each-other-talking-meeting-110846611.jpg?w=768"** },  {  **"id"**: **"user000003"**,  **"account"**: **"xiaoli"**,  **"email"**: **"xiaoli123@hotmail.com"**,  **"role"**: {  **"id"**: **"11911"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://images.pexels.com/photos/2625122/pexels-photo-2625122.jpeg?cs=srgb&dl=pexels-ali-pazani-2625122.jpg&fm=jpg"** },  {  **"id"**: **"user000004"**,  **"account"**: **"xiaoliu"**,  **"email"**: **"xiaoliubbb@hotmail.com"**,  **"role"**: {  **"id"**: **"1113"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://www.shutterstock.com/zh/image-photo/beautiful-blond-girl-light-eyes-posing-2363507355"** },  {  **"id"**: **"user000005"**,  **"account"**: **"xiaowei"**,  **"createDate"**: **null**,  **"email"**: **"xiaowei666@hotmail.com"**,  **"role"**: {  **"id"**: **"1114"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000006"**,  **"account"**: **"xiaguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo@hotmail.com"**,  **"role"**: {  **"id"**: **"1115"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000007"**,  **"account"**: **"xiagua"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo@hotmail.com"**,  **"role"**: {  **"id"**: **"1116"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000008"**,  **"account"**: **"xiacai"**,  **"createDate"**: **null**,  **"email"**: **"xiaocaiu@hotmail.com"**,  **"role"**: {  **"id"**: **"1117"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000009"**,  **"account"**: **"xiaguoli"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoli@hotmail.com"**,  **"role"**: {  **"id"**: **"1118"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000010"**,  **"account"**: **"daguo"**,  **"createDate"**: **null**,  **"email"**: **"daoguo@hotmail.com"**,  **"role"**: {  **"id"**: **"1119"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000011"**,  **"account"**: **"xiaguoz"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoz@hotmail.com"**,  **"role"**: {  **"id"**: **"1120"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000012"**,  **"account"**: **"xiakuo"**,  **"createDate"**: **null**,  **"email"**: **"xiaokuo@hotmail.com"**,  **"role"**: {  **"id"**: **"1121"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000013"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1122"**,  **"name"**: **"普通管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000014"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1123"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000015"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000016"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1211"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"5user0000017"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000018"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000019"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11110"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000020"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"11112"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000021"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11011"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000022"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000023"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000024"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"31111"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000025"**,  **"account"**: **"xiaguoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"41111"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000026"**,  **"account"**: **"xiaxiaoguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"71111"**,  **"name"**: **"用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** }  ] } |

### 所以我的代码和老师的不一样，不过由有一点是对的，删除数据后需要重新请求数据来渲染页面

### 我把users.js修改了一下

|  |
| --- |
| **import *request* from "../utils/request"**;  **export let** *getUserApi* = () => ***request***.get(**"/users"**) **export let** *delUserApi* = (id) =>{  ***request***.delete(**`/users/**${id}**`**).then(r => ***console***.log(r))  **return** id } |

### 然后把UsersPage.jsx的代码也修改了一下

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*delUserApi*, *getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**;   **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**a onClick=**{()=>delUser(record.**id**)}>删除</**a**>  </**Space**>  )  }  ,  },  ];   **let** [usersData,setUsersData] = *useState*([])  *useEffect*(()=>{  getUsers()  },[])   *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  **return** (  <>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

#### 功能基本上正常

## 6）给删除按钮的处理函数添加一个确认功能，不能一点就删除，这个确认框ant design也有提供，我们可以到这个网站里面取找一个自己喜欢的拷贝下来慢慢修改，我们可以现在一个Popconfirm组件来实现功能，然后我们把UsersPage.jsx修改如下

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*delUserApi*, *getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**; **import** { ***Button***, ***message***, ***Popconfirm*** } **from 'antd'**;  **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   **let** [usersData,setUsersData] = *useState*([])  *useEffect*(()=>{  getUsers()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  **return** (  <>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |
|  |

## 7）需要添加新增和修改，第一件事情，需要把我们是json-server启动：

### 在终端定位到E:\code\react-projects\React-fam-bucketdemo\public 然后执行json-server --watch data.json

新增功能不需要另外搞一个页面，只需要一个小弹窗。我们在页面上面新建一个按钮，点击按弹出弹窗，在按钮和表格直接添加一个Divider分割线组件

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*delUserApi*, *getUserApi*} **from "../../../api/users"**;  *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm*** } **from 'antd'**;  **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   **let** [usersData,setUsersData] = *useState*([])  *useEffect*(()=>{  getUsers()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  **return** (  <>  <**Button type="primary"**>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |

## 由于下面的新增需要由密码一栏。所以我们修改邮箱data.json,内容如下

|  |
| --- |
| {  **"users"**: [  {  **"id"**: **"user000001"**,  **"account"**: **"xiaozhang"**,  **"password"**: **"123"**,  **"email"**: **"xiaozhang123@hotmail.com"**,  **"role"**: {  **"id"**: **"1112"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://thumbs.dreamstime.com/b/delighted-nice-women-talking-meeting-friends-delighted-nice-positive-women-looking-each-other-talking-meeting-110846611.jpg?w=768"** },  {  **"id"**: **"user000002"**,  **"account"**: **"xiaoli"**,  **"password"**: **"123"**,  **"email"**: **"xiaoli123@hotmail.com"**,  **"role"**: {  **"id"**: **"11911"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://images.pexels.com/photos/2625122/pexels-photo-2625122.jpeg?cs=srgb&dl=pexels-ali-pazani-2625122.jpg&fm=jpg"** },  {  **"id"**: **"user000003"**,  **"account"**: **"ron"**,  **"password"**: **"123"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1122"**,  **"name"**: **"普通管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000004"**,  **"account"**: **"xiaxiaoguo"**,  **"password"**: **"123"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1123"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000005"**,  **"account"**: **"xiaguolu"**,  **"password"**: **"1233"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000006"**,  **"account"**: **"caixiaoguo"**,  **"password"**: **"12345"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1211"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000007"**,  **"account"**: **"xiaguocai"**,  **"password"**: **"4123"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000008"**,  **"account"**: **"lucy"**,  **"password"**: **"12223"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000009"**,  **"account"**: **"ben"**,  **"password"**: **"12003"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11110"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000010"**,  **"account"**: **"freddy"**,  **"password"**: **"12350"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"11112"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000011"**,  **"account"**: **"raymond"**,  **"password"**: **"123456"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11011"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000012"**,  **"account"**: **"john"**,  **"password"**: **"10123"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000013"**,  **"account"**: **"william"**,  **"password"**: **"12355"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000014"**,  **"account"**: **"nancy"**,  **"password"**: **"12456"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000015"**,  **"account"**: **"liliu"**,  **"password"**: **"11233"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000016"**,  **"account"**: **"mable"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000017"**,  **"account"**: **"stephany"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000018"**,  **"account"**: **"miky"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000019"**,  **"account"**: **"tamica"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000020"**,  **"account"**: **"luola"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000021"**,  **"account"**: **"paulin"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000022"**,  **"account"**: **"lulu"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000023"**,  **"account"**: **"sammy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000024"**,  **"account"**: **"ken"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000025"**,  **"account"**: **"rongyu"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000026"**,  **"account"**: **"brandy"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000027"**,  **"account"**: **"andy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000028"**,  **"account"**: **"anita"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000029"**,  **"account"**: **"afred"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000030"**,  **"account"**: **"derrick"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000031"**,  **"account"**: **"candy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** }  ] } |

## 8）我们到ant design网站找一个弹窗组件，我们使用modal组件，然后我们把它修改一下，改为我们需要的样子。然后我们需要在modal组件里面嵌套一个form表单，我们去ant design找一个，我们这里把表单的onFinish和onFinishFailed事件处理函数删除，因为我们这里不使用表单的提交功能，我们使用弹窗按钮的点击事件，然后需要从后端请求角色数据，我们先创建一个roles.js来封装相关的api，需要一个roles接口，我们把data.json修改一下添加roles接口

|  |
| --- |
| {  **"users"**: [  {  **"id"**: **"user000001"**,  **"account"**: **"xiaozhang"**,  **"password"**: **"123"**,  **"email"**: **"xiaozhang123@hotmail.com"**,  **"role"**: {  **"id"**: **"1112"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://thumbs.dreamstime.com/b/delighted-nice-women-talking-meeting-friends-delighted-nice-positive-women-looking-each-other-talking-meeting-110846611.jpg?w=768"** },  {  **"id"**: **"user000002"**,  **"account"**: **"xiaoli"**,  **"password"**: **"123"**,  **"email"**: **"xiaoli123@hotmail.com"**,  **"role"**: {  **"id"**: **"11911"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://images.pexels.com/photos/2625122/pexels-photo-2625122.jpeg?cs=srgb&dl=pexels-ali-pazani-2625122.jpg&fm=jpg"** },  {  **"id"**: **"user000003"**,  **"account"**: **"xiaguoliu"**,  **"password"**: **"123"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1122"**,  **"name"**: **"普通管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000004"**,  **"account"**: **"xiaxiaoguo"**,  **"password"**: **"123"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1123"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000005"**,  **"account"**: **"xiaguolu"**,  **"password"**: **"1233"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000006"**,  **"account"**: **"caixiaoguo"**,  **"password"**: **"12345"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1211"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000007"**,  **"account"**: **"xiaguocai"**,  **"password"**: **"4123"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000008"**,  **"account"**: **"lucy"**,  **"password"**: **"12223"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"1111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000009"**,  **"account"**: **"ben"**,  **"password"**: **"12003"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11110"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000010"**,  **"account"**: **"freddy"**,  **"password"**: **"12350"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"11112"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000011"**,  **"account"**: **"raymond"**,  **"password"**: **"123456"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"11011"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000012"**,  **"account"**: **"john"**,  **"password"**: **"10123"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000013"**,  **"account"**: **"william"**,  **"password"**: **"12355"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000014"**,  **"account"**: **"nancy"**,  **"password"**: **"12456"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000015"**,  **"account"**: **"liliu"**,  **"password"**: **"11233"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000016"**,  **"account"**: **"mable"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000017"**,  **"account"**: **"stephany"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000018"**,  **"account"**: **"miky"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000019"**,  **"account"**: **"tamica"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000020"**,  **"account"**: **"luola"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000021"**,  **"account"**: **"paulin"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000022"**,  **"account"**: **"lulu"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000023"**,  **"account"**: **"sammy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000024"**,  **"account"**: **"ken"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000025"**,  **"account"**: **"rongyu"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000026"**,  **"account"**: **"brandy"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000027"**,  **"account"**: **"andy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000028"**,  **"account"**: **"anita"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000029"**,  **"account"**: **"afred"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000030"**,  **"account"**: **"derrick"**,  **"password"**: **"12378"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"12211"**,  **"name"**: **"网络管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000031"**,  **"account"**: **"candy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"111111"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** }  ],  **"roles"**: [  {  **"id"**: **"role0001"**,  **"menus"**: [**"/home"**,**"/login"**],  **"name"**: **"普通用户"**,  **"state"**: 1  },  {  **"id"**: **"role0002"**,  **"menus"**: [**"/home"**,**"/login"**],  **"name"**: **"vip用户"**,  **"state"**: 1  },  {  **"id"**: **"role0003"**,  **"menus"**: [**"/home"**,**"/login"**,**"/charts"**,**"charts/bar"**,**"charts/line"**,**"charts/pie"**],  **"name"**: **"统计用户"**,  **"state"**: 1  },  {  **"id"**: **"role0004"**,  **"menus"**: [**"/home"**,**"/login"**,**"/products/list"**,**"/products/category"**],  **"name"**: **"普通管理员"**,  **"state"**: 1  },  {  **"id"**: **"role0005"**,  **"menus"**: [**"/home"**,**"/login"**,**"/products"**,**"/role"**,**"/products/list"**,**"products/category"**],  **"name"**: **"超级管理员"**,  **"state"**: 1  }  ] } |

### 注意：老师的新增接口和我的不一样，roles不是写死的，是从后端请求回来的

## 9)在roles.js中封装角色请求api

### src/api/roles.js

|  |
| --- |
| **import *request* from "../utils/request"**;  **export let** *getRolesApi* = ()=>***request***.get(**"/roles"**) |

## 10)测试获取roles的数据是否成功，在UsersPage.jsx组件中添加代码

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   *//新增用户的弹出组件相关代码* **const** [isModalOpen, setIsModalOpen] = *useState*(**false**);  **const** showModal = () => {  setIsModalOpen(**true**);  };  **const** addUsers = () => {  setIsModalOpen(**false**);  };  **const** handleCancel = () => {  setIsModalOpen(**false**);  };    **let** [usersData,setUsersData] = *useState*([])  **let** [rolesData,setRolesData] = *useState*([])  *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  *// console.log(roles)* setRolesData([...roles])  ***console***.log(rolesData)  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  **const** roleChange = (value) => {  ***console***.log(**`selected** ${value}**`**);  };  **return** (  <>  <**Modal title="新增用户" open=**{isModalOpen} **onOk=**{addUsers} **onCancel=**{handleCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}}>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select   onChange=**{roleChange}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 我们发现数据是获取到了，没有问题，但是下拉列表里面显示不出来，因为我们的数据格式的问题，我们需要使用一个fileNames属性，传递一个对象，指明显示方式

|  |
| --- |
|  |

#### 然后下拉列表的角色数据就可以显示了

|  |
| --- |
|  |

## 11）实现当我们输入完表单信息点击弹窗的ok按钮时的数据处理，我们需要先封装一个api来新增数据，打开users.js文件，添加addUserApi

|  |
| --- |
| **import *request* from "../utils/request"**;  **export let** *getUserApi* = () => ***request***.get(**"/users"**) **export let** *delUserApi* = (id) =>{  ***request***.delete(**`/users/**${id}**`**).then(r => ***console***.log(r))  **return** id }  **export let** *addUserApi* = (data)=>***request***.post(**"/users"**,data) |

## 12）由于我的不是真正后端，我需要在表单里面添加id一栏，

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   *//新增用户的弹出组件相关代码* **const** [isModalOpen, setIsModalOpen] = *useState*(**false**);  **const** showModal = () => {  setIsModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsModalOpen(**false**);  };  **const** handleCancel = () => {  setIsModalOpen(**false**);  };    **let** [usersData,setUsersData] = *useState*([])  **let** [rolesData,setRolesData] = *useState*([])  **let** [addData,setAddData] = *useState*({})    *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **let** onValuesChange = (changedValues,allValues) =>{  setAddData(allValues)  }   **return** (  <>  <**Modal title="新增用户" open=**{isModalOpen} **onOk=**{addUsers} **onCancel=**{handleCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onValuesChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |
|  |

注意，由于我的不是真正后端，角色属性又是一个对象，所以我得想办法封装数据 ，没有办法，json-server不会自动封装数据，需要修改onValuesChange函数的代码，如果后端可以自动封装对象，很多代码都不用写，不过学习没有办法

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**; **import** {*all*} **from "axios"**; **import *request* from "../../../utils/request"**;   **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   *//新增用户的弹出组件相关代码* **const** [isModalOpen, setIsModalOpen] = *useState*(**false**);  **const** showModal = () => {  setIsModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsModalOpen(**false**);  getUsers()  };  **const** handleCancel = () => {  setIsModalOpen(**false**);  };    **let** [usersData,setUsersData] = *useState*([])  **let** [rolesData,setRolesData] = *useState*([])  **let** [addData,setAddData] = *useState*({})    *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **let** onValuesChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   **return** (  <>  <**Modal title="新增用户" open=**{isModalOpen} **onOk=**{addUsers} **onCancel=**{handleCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onValuesChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

## 新增的功能完成了，下面我们来做修改功能

## 13）把弹窗组件复制一份，作为修改的弹窗，其实，我们可以把弹窗组件抽取处理单独作为一个文件也是可以的，而且更好。以后我们慢慢优化，注意上面的代码也要修改一下否则有问题，必须保证两个弹窗是由不同的打开状态来控制的。根据不同的状态打开对应的弹窗

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a**>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];   *//新增用户的弹出组件相关代码* **const** [isAddModalOpen, setIsAddModalOpen] = *useState*(**false**);  *//修改用户的弹窗状态数据* **const** [isUpdateModalOpen, setIsUpdateModalOpen] = *useState*(**false**);  **const** showAddModal = () => {  setIsAddModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsAddModalOpen(**false**);  getUsers()  };  **const** handleAddCancel = () => {  setIsAddModalOpen(**false**);  };  **const** showUpdateModal = () => {  setIsUpdateModalOpen(**true**);  };  *//修改用户数据功能* **let** updateUsers = () =>{  setIsUpdateModalOpen(**false**)  }  **let** handleUpdateCancel = () =>{  setIsUpdateModalOpen(**false**)  }  *//收集表单数据并且设置到需要保存的数据状态中* **let** onAddChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   *//收集表单数据并且保存到需要修改的数据状态功能* **let** onUpdateChange = (changedValues,allValues) =>{   }   **let** [usersData,setUsersData] = *useState*([])  **let** [rolesData,setRolesData] = *useState*([])  **let** [addData,setAddData] = *useState*({})    *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **return** (  <>  {*/\*新增用户弹窗\*/*}  <**Modal title="新增用户" open=**{isAddModalOpen} **onOk=**{addUsers} **onCancel=**{handleAddCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onAddChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  {*/\*修改用户弹窗\*/*}  <**Modal title="修改用户" open=**{isUpdateModalOpen} **onOk=**{updateUsers} **onCancel=**{handleUpdateCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onUpdateChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showAddModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

## 14）给修改超链接添加点击事件处理

|  |
| --- |
|  |

## 15）用useState函数新建一个updateData状态，用来保存需要修改的数据，我们把代码修改一下

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  *//状态数据  //保存所有用户状态数据* **let** [usersData,setUsersData] = *useState*([])  *//保存所有角色状态数据* **let** [rolesData,setRolesData] = *useState*([])  *//保存需要新增的数据的状态* **let** [addData,setAddData] = *useState*({})  *//保存需要修改的数据的状态* **let** [updateData,setUpdateData] = *useState*({})  *//控制新增用户的弹窗显示隐藏的状态* **const** [isAddModalOpen, setIsAddModalOpen] = *useState*(**false**);  *//控制修改用户的弹窗状显示隐藏态数据* **const** [isUpdateModalOpen, setIsUpdateModalOpen] = *useState*(**false**);  **const** showAddModal = () => {  setIsAddModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsAddModalOpen(**false**);  getUsers()  };  **const** handleAddCancel = () => {  setIsAddModalOpen(**false**);  };  **const** showUpdateModal = () => {  setIsUpdateModalOpen(**true**);  };  *//获取弹窗数据，打开弹窗，修改数据* **let** getUpdateUser = (updateData) =>{  setUpdateData(updateData)  setIsUpdateModalOpen(**true**)  }  *//修改用户数据功能* **let** updateUsers = () =>{  setIsUpdateModalOpen(**false**)  }  **let** handleUpdateCancel = () =>{  setIsUpdateModalOpen(**false**)  }  *//收集表单数据并且设置到需要保存的数据状态中* **let** onAddChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   *//收集表单数据并且保存到需要修改的数据状态功能* **let** onUpdateChange = (changedValues,allValues) =>{   }   *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a onClick=**{()=>getUpdateUser(record)}>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];     **return** (  <>  {*/\*新增用户弹窗\*/*}  <**Modal title="新增用户" open=**{isAddModalOpen} **onOk=**{addUsers} **onCancel=**{handleAddCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onAddChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  {*/\*修改用户弹窗\*/*}  <**Modal title="修改用户" open=**{isUpdateModalOpen} **onOk=**{updateUsers} **onCancel=**{handleUpdateCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onUpdateChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showAddModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

## 16）我们已经获取到了选中的数据，现在需要把它设置为对应表单的默认值，使用表单的initialValues属性其实是可以的只不过我们不能用setState来更新它。需要使用setFieldValue来更新，注意我的代码和老师的由一点点不一样，setFieldValue的用法如下，我们把用户id禁用，因为修改用户信息不需要修改用户的id

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  *//状态数据  //保存所有用户状态数据* **let** [usersData,setUsersData] = *useState*([])  *//保存所有角色状态数据* **let** [rolesData,setRolesData] = *useState*([])  *//保存需要新增的数据的状态* **let** [addData,setAddData] = *useState*({})  *//保存需要修改的数据的状态* **let** [updateData,setUpdateData] = *useState*({})  *//控制新增用户的弹窗显示隐藏的状态* **const** [isAddModalOpen, setIsAddModalOpen] = *useState*(**false**);  *//控制修改用户的弹窗状显示隐藏态数据* **const** [isUpdateModalOpen, setIsUpdateModalOpen] = *useState*(**false**);  *//更新表单数据需要的* **const** [form] = ***Form***.*useForm*()   **const** showAddModal = () => {  setIsAddModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsAddModalOpen(**false**);  getUsers()  };  **const** handleAddCancel = () => {  setIsAddModalOpen(**false**);  };  **const** showUpdateModal = () => {  setIsUpdateModalOpen(**true**);  };  *//获取弹窗数据，打开弹窗，修改数据* **let** getUpdateUser = (updateData) =>{  setUpdateData(updateData)*//虽然改变了数据，但是表单不会更新  //更新表单默认值* form.**setFieldsValue**({*//注意：这里不能直接私有...了扩展updateData* **id**:updateData.**id**,  **account**:updateData.**account**,  **password**:updateData.**password**,  **email**:updateData.**email**,  **role**:updateData.**role**.**id** })  setIsUpdateModalOpen(**true**)  }  *//修改用户数据功能* **let** updateUsers = () =>{  setIsUpdateModalOpen(**false**)  }  **let** handleUpdateCancel = () =>{  setIsUpdateModalOpen(**false**)  }  *//收集表单数据并且设置到需要保存的数据状态中* **let** onAddChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   *//用户修改表单数据时触发这个函数* **let** onUpdateChange = (changedValues,allValues) =>{  **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setUpdateData(allValues)  }   *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a onClick=**{()=>getUpdateUser(record)}>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];     **return** (  <>  {*/\*新增用户弹窗\*/*}  <**Modal title="新增用户" open=**{isAddModalOpen} **onOk=**{addUsers} **onCancel=**{handleAddCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onAddChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  {*/\*修改用户弹窗\*/*}  <**Modal title="修改用户" open=**{isUpdateModalOpen} **onOk=**{updateUsers} **onCancel=**{handleUpdateCancel}>  <**Form form=**{form} **labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}}  **onValuesChange=**{onUpdateChange} **initialValues=**{updateData}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input disabled="disabled"** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showAddModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |

### 使用setFieldValue的步骤，

#### 1.创建一个form变量 const [form] = Form.useForm(),

#### 2.在antd的对应Form表单上面添加一个属性form = {form}

#### 3.然后使用form.setFieldValue({ })

## 17)上面只是完成了打开修改对话框并且显示回显，下面我们来做数据修改，请参考updateUsers函数的代码，

### UsersPage.jsx

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  *//状态数据  //保存所有用户状态数据* **let** [usersData,setUsersData] = *useState*([])  *//保存所有角色状态数据* **let** [rolesData,setRolesData] = *useState*([])  *//保存需要新增的数据的状态* **let** [addData,setAddData] = *useState*({})  *//保存需要修改的数据的状态* **let** [updateData,setUpdateData] = *useState*({})  *//控制新增用户的弹窗显示隐藏的状态* **const** [isAddModalOpen, setIsAddModalOpen] = *useState*(**false**);  *//控制修改用户的弹窗状显示隐藏态数据* **const** [isUpdateModalOpen, setIsUpdateModalOpen] = *useState*(**false**);  *//更新表单数据需要的* **const** [form] = ***Form***.*useForm*()   **const** showAddModal = () => {  setIsAddModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsAddModalOpen(**false**);  getUsers()  };  **const** handleAddCancel = () => {  setIsAddModalOpen(**false**);  };  **const** showUpdateModal = () => {  setIsUpdateModalOpen(**true**);  };  *//获取弹窗数据，打开弹窗，修改数据* **let** getUpdateUser = (updateData) =>{  setUpdateData(updateData)*//虽然改变了数据，但是表单不会更新  //更新表单默认值* form.**setFieldsValue**({*//注意：这里不能直接私有...了扩展updateData* **id**:updateData.**id**,  **account**:updateData.**account**,  **password**:updateData.**password**,  **email**:updateData.**email**,  **role**:updateData.**role**.**id** })  setIsUpdateModalOpen(**true**)  }  *//修改用户数据功能* **let** updateUsers = () =>{   setIsUpdateModalOpen(**false**)  }  **let** handleUpdateCancel = () =>{  setIsUpdateModalOpen(**false**)  }  *//收集表单数据并且设置到需要保存的数据状态中* **let** onAddChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   *//用户修改表单数据时触发这个函数* **let** onUpdateChange = (changedValues,allValues) =>{  **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setUpdateData(allValues)  }   *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a onClick=**{()=>getUpdateUser(record)}>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.id)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];     **return** (  <>  {*/\*新增用户弹窗\*/*}  <**Modal title="新增用户" open=**{isAddModalOpen} **onOk=**{addUsers} **onCancel=**{handleAddCancel}>  <**Form labelCol=**{{span: 4,}} **wrapperCol=**{{span: 18,}} **style=**{{maxWidth: 600,}} **onValuesChange=**{onAddChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  required: **true**,  message: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  required: **true**,  message: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  required: **true**,  message: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  required: **true**,  message: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{label:**'name'**,value:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  {*/\*修改用户弹窗\*/*}  <**Modal title="修改用户" open=**{isUpdateModalOpen} **onOk=**{updateUsers} **onCancel=**{handleUpdateCancel}>  <**Form form=**{form} **labelCol=**{{span: 4,}} **wrapperCol=**{{span: 18,}} **style=**{{maxWidth: 600,}}  **onValuesChange=**{onUpdateChange} **initialValues=**{updateData}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  required: **true**,  message: **'请输入用户id!'**,  },  ]}  >  <**Input disabled="disabled"** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  required: **true**,  message: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  required: **true**,  message: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  required: **true**,  message: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{label:**'name'**,value:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showAddModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** UsersPage; |

## 18)发送put请求实现数据修改，我们在users.js里面新增一个修改数据请求

|  |
| --- |
| **import *request* from "../utils/request"**;  *//获取使用用户列表* **export let** *getUserApi* = () => ***request***.get(**"/users"**) *//删除用户* **export let** *delUserApi* = (id) =>{  ***request***.delete(**`/users/**${id}**`**).then(r => ***console***.log(r))  **return** id } *//新增用户* **export let** *addUserApi* = (data)=>***request***.post(**"/users"**,data) *//修改用户信息* **export let** *updateUserApi* = (id,data) =>***request***.put(**`/users/**${id}**`**,data) |

## 19）然后在updateUser方法里面调用我们的修改接口，完成数据修改

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** {*addUserApi*, *delUserApi*, *getUserApi*, *updateUserApi*} **from "../../../api/users"**; **import** {*getRolesApi*} **from "../../../api/roles"**; *//引入表格组件* **import** { ***Space***, ***Table***, ***Tag***, ***Modal***,***Select*** } **from 'antd'**; **import** { ***Button***, ***Divider***, ***Popconfirm***,***Form***,***Input*** } **from 'antd'**;   **const** *UsersPage* = () => {  *//状态数据  //保存所有用户状态数据* **let** [usersData,setUsersData] = *useState*([])  *//保存所有角色状态数据* **let** [rolesData,setRolesData] = *useState*([])  *//保存需要新增的数据的状态* **let** [addData,setAddData] = *useState*({})  *//保存需要修改的数据的状态* **let** [updateData,setUpdateData] = *useState*({})  *//控制新增用户的弹窗显示隐藏的状态* **const** [isAddModalOpen, setIsAddModalOpen] = *useState*(**false**);  *//控制修改用户的弹窗状显示隐藏态数据* **const** [isUpdateModalOpen, setIsUpdateModalOpen] = *useState*(**false**);  *//更新表单数据需要的* **const** [form] = ***Form***.*useForm*()   **const** showAddModal = () => {  setIsAddModalOpen(**true**);  };  **const** addUsers = **async** () => {   *//发送post请求新增数据* **let** res = **await** *addUserApi*(addData)  ***console***.log(**"result:"**,res)  setIsAddModalOpen(**false**);  getUsers()  };  **const** handleAddCancel = () => {  setIsAddModalOpen(**false**);  };  **const** showUpdateModal = () => {  setIsUpdateModalOpen(**true**);  };  *//获取弹窗数据，打开弹窗，修改数据* **let** getUpdateUser = (updateData) =>{  setUpdateData(updateData)*//虽然改变了数据，但是表单不会更新  //更新表单默认值* form.**setFieldsValue**({*//注意：这里不能直接私有...了扩展updateData* **id**:updateData.**id**,  **account**:updateData.**account**,  **password**:updateData.**password**,  **email**:updateData.**email**,  **role**:updateData.**role**.**id** })  setIsUpdateModalOpen(**true**)  }  *//修改用户数据功能* **let** updateUsers = () =>{  ***console***.log(**"data ready:"**,updateData)  *updateUserApi*(updateData.**id**,updateData)  getUsers()  setIsUpdateModalOpen(**false**)  }  **let** handleUpdateCancel = () =>{  setIsUpdateModalOpen(**false**)  }  *//收集表单数据并且设置到需要保存的数据状态中* **let** onAddChange = (changedValues,allValues) =>{  *// console.log("rolesdata:",rolesData)* **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setAddData(allValues)  }   *//用户修改表单数据时触发这个函数* **let** onUpdateChange = (changedValues,allValues) =>{  **if**(allValues.**role**){  **let** id = allValues.**role  let** obj = rolesData.filter(item=>item.**id** == id)[0]  *// console.log("filter result:",obj)* **let** roleObj = {  **id**:id,  **name**:obj.**name** }  allValues.**role** =roleObj  }  ***console***.log(allValues)  setUpdateData(allValues)  }   *useEffect*(()=>{  getUsers()  getRoles()  },[])    *//获取所有用户* **let** getUsers = **async** () =>{  **let** res = **await** *getUserApi*()  *//如果你请求的是真正的后端，需要用res.code来判断是否请求成功，1是成功，0是失败* **if**(res){  ***console***.log(res)  setUsersData([...res]) *//这个res不是数组，只是一个类数组对象，需要利用扩展操作符给数组赋值* }   }  *//获取所有角色* **let** getRoles = **async** () =>{  **let** roles = **await** *getRolesApi*()  ***console***.log(roles)  setRolesData([...roles])  }  *//删除用户* **let** delUser = **async** (id) =>{  **let** res = *delUserApi*(id)  ***console***.log(res)  **await** getUsers()   }  *// const roleChange = (value) => {  // console.log(`selected ${value}`);  // };* **const** columns = [  {  **title**: **'账号'**,  **dataIndex**: **'account'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'邮箱'**,  **dataIndex**: **'email'**,  },  {  **title**: **'角色'**,  **dataIndex**: [**'role'**,**'name'**],  render:(text)=>text===**"超级管理员"** ? <**Tag color="volcano"**>{text}</**Tag**> : <**span**>{text}</**span**>  },   {  **title**: **'操作'**,  **key**: **'action'**,  render: (\_, record) => {  **return** (  <**Space size="middle"**>  <**a onClick=**{()=>getUpdateUser(record)}>修改</**a**>  <**Popconfirm  title="删除确认"  description="删除不可恢复，你确定删除吗?"  onConfirm=**{()=>delUser(record.**id**)}  **cancelText="取消"  okText="确认"** >  <**a**>删除</**a**>  </**Popconfirm**>   </**Space**>  )  }  ,  },  ];     **return** (  <>  {*/\*新增用户弹窗\*/*}  <**Modal title="新增用户" open=**{isAddModalOpen} **onOk=**{addUsers} **onCancel=**{handleAddCancel}>  <**Form labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}} **onValuesChange=**{onAddChange}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  {*/\*修改用户弹窗\*/*}  <**Modal title="修改用户" open=**{isUpdateModalOpen} **onOk=**{updateUsers} **onCancel=**{handleUpdateCancel}>  <**Form form=**{form} **labelCol=**{{**span**: 4,}} **wrapperCol=**{{**span**: 18,}} **style=**{{**maxWidth**: 600,}}  **onValuesChange=**{onUpdateChange} **initialValues=**{updateData}>  <**Form.Item  label="用户id"  name="id"  rules=**{[  {  **required**: **true**,  **message**: **'请输入用户id!'**,  },  ]}  >  <**Input disabled="disabled"** />  </**Form.Item**>  <**Form.Item  label="账号"  name="account"  rules=**{[  {  **required**: **true**,  **message**: **'请输入账号'**,  },  ]}  >  <**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="邮箱"  name="email"  rules=**{[  {  **required**: **true**,  **message**: **'请输入邮箱!'**,  },  ]}  >  <**Input** />  </**Form.Item**>  <**Form.Item  label="角色"  name="role"  rules=**{[  {  **required**: **true**,  **message**: **'请选择角色!'**,  },  ]}  >  <**Select fieldNames=**{{**label**:**'name'**,**value**:**'id'**}}  **options=**{rolesData}  />  </**Form.Item**>   </**Form**>  </**Modal**>  <**Button type="primary" onClick=**{showAddModal}>新增</**Button**>  <**Divider**/>  <**Table columns=**{columns} **dataSource=**{usersData} **rowKey="id"**/>;  </>  ); };  **export default** *UsersPage*; |

### 效果

|  |
| --- |
|  |
|  |

# 至此，使用用户功能完成