# react全家桶学习笔记1：[学习笔记1-路由基础配置](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](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](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](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：axios配置网络请求

# 还是React-fam-bucketdemo项目，我们来做用户管理功能

## 1.网络请求的配置

### 1）安装axios

因为我们这个项目是用yarn创建的，所以我们用yarn add axios来安装

### 2）基本使用，用几种用法如：直接axios({...}),这里我们使用它的别名语法

安装完成axios后，使用yarn dev启动项目。

|  |  |
| --- | --- |
| 请求方式 | axios方法 |
| get | axios.get(“请求路径”, { params:{请求参数}}) |
| post | axios.post(“请求路径”, {请求参数}) |
| put | axios.put |
| delete | axios.delete |

## 2.回到项目，我们打开UsersPage组件文件，在这里使用axios发送请求

### 由于我没有后端接口，我使用json文件夹+json-server来模拟后端接口

### 在项目的public项目下面新建一个data.json

|  |
| --- |
| {  **"users"**:[  {  **"\_id"**: **"5fb3bf5db1550000370000b3"**,  **"account"**: **"xiaowang"**,  **"createDate"**: **null**,  **"email"**: **"xiaowang123@hotmail.com"**,  **"imgUrl"**:**"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"\_id"**: **"5fb3bf5db1550000370000c3"**,  **"account"**: **"xiaozhang"**,  **"createDate"**: **null**,  **"email"**: **"xiaozhang123@hotmail.com"**,  **"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"**: **"5fb3bf5db1550000370000=f3"**,  **"account"**: **"xiaoli"**,  **"createDate"**: **null**,  **"email"**: **"xiaoli123@hotmail.com"**,  **"imgUrl"**:**"https://images.pexels.com/photos/2625122/pexels-photo-2625122.jpeg?cs=srgb&dl=pexels-ali-pazani-2625122.jpg&fm=jpg"** },  {  **"\_id"**: **"5fb3bf5db1550000370000b3"**,  **"account"**: **"xiaoliu"**,  **"createDate"**: **null**,  **"email"**: **"xiaoliubbb@hotmail.com"**,  **"imgUrl"**:**"https://www.shutterstock.com/zh/image-photo/beautiful-blond-girl-light-eyes-posing-2363507355"** },  {  **"\_id"**: **"5fb3bf5db1550000370000b3"**,  **"account"**: **"xiaowei"**,  **"createDate"**: **null**,  **"email"**: **"xiaowei666@hotmail.com"**,  **"imgUrl"**:**"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"\_id"**: **"5fb3bf5db1550000370000b3"**,  **"account"**: **"xiaguo"**,  **"createDate"**: **null**,  **"email"**: **"xiaoguo@hotmail.com"**,  **"imgUrl"**:**"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** }  ] } |

### 打开一个终端定位到项目的public下面，然后使用命令：json-server --watch data.json

### 然后我们的访问路径是：http://localhost:3000/users/

### 用axios模拟向后端发送请求的代码如下

#### UsersPages.jsx,注意需要使用useEffect钩子函数来模拟componentDidMount生命周期

|  |
| --- |
| **import *React***, {*useEffect*} **from 'react'**; **import *axios* from 'axios'  const** *UsersPage* = () => {  *useEffect*(()=>{ //模拟生命周期函数  getUsers()  },[])  **let** getUsers = **async** () =>{  **let** res = **await *axios***.get(**"http://localhost:3000/users"**)  ***console***.log(res)  }  **return** (  <**div**>  users  </**div**>  ); };  **export default** *UsersPage*; |

#### 效果：

|  |
| --- |
|  |

### 在实际开发中，上面的写法不太好，我们需要把它进行封装

## 3.axios封装

### 1）先在项目的src文件夹下面创建一个utils文件夹，然后在里面新建一个request.js文件，在里面配置和封装axios，注意这里我们用axios的create方法把axios复制一份，我们的配置是在副本里面完成的，这样子做的好处是万一以后有一些特殊的请求不能用我们的axios副本来做，我们还可以用axios来做，相当于把axios当作备用的

|  |
| --- |
| **import *axios* from "axios"**;  *//创建一个请求对象* **let *request*** = ***axios***.create({  **baseURL**:**"http://localhost:3000"**,  **timeout**:5000 })  *//配置响应拦截器* ***request***.**interceptors**.**response**.use(res => res.**data**)  **export default *request*** |

### 2）然后上面的请求代码就响应做相应的修改，注意：我没有老师的后端接口，我的路径和老师的是不一样的

|  |
| --- |
| **import *React***, {*useEffect*} **from 'react'**; **import *axios* from 'axios' import *request* from "../../../utils/request"**;  **const** *UsersPage* = () => {  *useEffect*(()=>{  getUsers()  },[])  **let** getUsers = **async** () =>{  *// let res = await axios.get("http://localhost:3000/users")  // console.log(res)*  *//在request里面已经把基础路径设置为*[*http://localhost:3000*](http://localhost:3000)*了，所以可以直接写/users* **let** data = **await *request***.get(**"/users"**)  ***console***.log(data)  }  **return** (  <**div**>  users  </**div**>  ); };  **export default** *UsersPage*; |

#### 效果：拿到数据了

|  |
| --- |
|  |

## 4.api封装

### 在src目录下面新建一个api文件夹，然后新建一个users.js,内容如下

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

### 然后，在UsersPages.jsx中调用api，注意请求我们的json-server和请求真正的后端是有点不一样的，需要注意一下，它可能没有code，请求后端有code

|  |
| --- |
|  |

#### 我们的代码如下

|  |
| --- |
| **import *React***, {*useEffect*} **from 'react'**; **import *axios* from 'axios' import *request* from "../../../utils/request"**; **import** {*getUserApi*} **from "../../../api/users"**;  **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** (  <**div**>  users  </**div**>  ); };  **export default** *UsersPage*; |

#### 注意：请求后端会有包装2次data，json-server只有一次

## 5.拿到数据后，我们可以到ant design去找一个表格组件来渲染数据