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](react全家桶学习笔记6-antd用户列表渲染.docx)

# react全家桶学习笔记7：路由传参

# 还是React-fam-bucketdemo项目，我们现在来做路由传参

# 我们今天来完成商品管理功能，先来看商品列表

## 1）商品列表渲染，我们到ant design网站找另外一种表格，多学一点，他默认的代码是这样子的

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table*** } **from 'antd'**; **const** columns = [  {  **title**: **'Name'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'Age'**,  **dataIndex**: **'age'**,  },  {  **title**: **'Address'**,  **dataIndex**: **'address'**,  }, ]; **const** data = [  {  **key**: **'1'**,  **name**: **'John Brown'**,  **age**: 32,  **address**: **'New York No. 1 Lake Park'**,  },  {  **key**: **'2'**,  **name**: **'Jim Green'**,  **age**: 42,  **address**: **'London No. 1 Lake Park'**,  },  {  **key**: **'3'**,  **name**: **'Joe Black'**,  **age**: 32,  **address**: **'Sydney No. 1 Lake Park'**,  },  {  **key**: **'4'**,  **name**: **'Disabled User'**,  **age**: 99,  **address**: **'Sydney No. 1 Lake Park'**,  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  },  getCheckboxProps: (record) => ({  **disabled**: record.**name** === **'Disabled User'**,  *// Column configuration not to be checked* **name**: record.**name**,  }), };  **const** *ProductsList* = () => {  *//状态数据* **const** [selectionType, setSelectionType] = *useState*(**'checkbox'**);  **return** (  <**div**>  <**Radio.Group  onChange=**{({ **target**: { value } }) => {  setSelectionType(value);  }}  **value=**{selectionType}  >  <**Radio value="checkbox"**>Checkbox</**Radio**>  <**Radio value="radio"**>radio</**Radio**>  </**Radio.Group**>   <**Divider** />   <**Table  rowSelection=**{{  **type**: selectionType,  ...rowSelection,  }}  **columns=**{columns}  **dataSource=**{data}  />  </**div**>  ); };  **export default** *ProductsList*; |

## 2）把一些不需要的内容删除后，变为这样子

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table*** } **from 'antd'**; **const** columns = [  {  **title**: **'Name'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'Age'**,  **dataIndex**: **'age'**,  },  {  **title**: **'Address'**,  **dataIndex**: **'address'**,  }, ]; **const** data = [  {  **key**: **'1'**,  **name**: **'John Brown'**,  **age**: 32,  **address**: **'New York No. 1 Lake Park'**,  },  {  **key**: **'2'**,  **name**: **'Jim Green'**,  **age**: 42,  **address**: **'London No. 1 Lake Park'**,  },  {  **key**: **'3'**,  **name**: **'Joe Black'**,  **age**: 32,  **address**: **'Sydney No. 11 Lake Park'**,  },  {  **key**: **'4'**,  **name**: **'John'**,  **age**: 29,  **address**: **'Sydney No. 1 Lake Park'**,  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  } };  **const** *ProductsList* = () => {   **return** (  <**div**>  <**Table  rowSelection=**{{**type**: **'checkbox'**, ...rowSelection}}  **columns=**{columns}  **dataSource=**{data}  />  </**div**>  ); };  **export default** *ProductsList*; |

### 效果

|  |
| --- |
|  |

## 3）在api文件夹里面新一个products.js文件，添加商品管理相关的api封装，注意，我的接口和老师的不一样，我又需要自己编写json数据

### 老师的接口

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

### 1.我们在data.json文件里面添加goods路径，数据如下

|  |
| --- |
| {  **"users"**: [  {  **"id"**: **"user000001"**,  **"account"**: **"xiaozheng"**,  **"password"**: **"123"**,  **"email"**: **"xiaozheng123@hotmail.com"**,  **"role"**: {  **"id"**: **"role00002"**,  **"name"**: **"vip用户"** }  },  {  **"id"**: **"user000002"**,  **"account"**: **"laoli"**,  **"password"**: **"123"**,  **"email"**: **"laoli123@hotmail.com"**,  **"role"**: {  **"id"**: **"role00005"**,  **"name"**: **"超级管理员"** }  },  {  **"id"**: **"user000003"**,  **"account"**: **"xiaoxiaoliu"**,  **"password"**: **"1234"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00004"**,  **"name"**: **"普通管理员"** }  },  {  **"id"**: **"user000004"**,  **"account"**: **"xiaxiaoguo"**,  **"password"**: **"123"**,  **"email"**: **"xiaoguo2@hotmail.com"**,  **"role"**: {  **"id"**: **"role00002"**,  **"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"**: **"role00001"**,  **"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"**: **"role00001"**,  **"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"**: **"role00006"**,  **"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"**: **"role00001"**,  **"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"**: **"role00001"**,  **"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"**: **"role00001"**,  **"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"**: **"role00001"**,  **"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"**: **"role00007"**,  **"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"**: **"role00001"**,  **"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"**: **"role00007"**,  **"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"**: **"role00001"**,  **"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"**: **"role00007"**,  **"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"**: **"role00001"**,  **"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"**: **"role00001"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000021"**,  **"account"**: **"paulin"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"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"**: **"role00006"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000023"**,  **"account"**: **"sammy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"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"**: **"role00006"**,  **"name"**: **"数据管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000025"**,  **"account"**: **"rongyu"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"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"**: **"role00005"**,  **"name"**: **"超级管理员"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000027"**,  **"account"**: **"andy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"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"**: **"role00002"**,  **"name"**: **"vip用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000029"**,  **"account"**: **"afred"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"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"**: **"role00001"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user0000031"**,  **"account"**: **"candy"**,  **"password"**: **"123666"**,  **"email"**: **"xiaoguoliu@hotmail.com"**,  **"role"**: {  **"id"**: **"role00001"**,  **"name"**: **"普通用户"** },  **"imgUrl"**: **"https://i.pinimg.com/736x/c9/21/42/c92142f2ef427934cef9c66bafdc6191.jpg"** },  {  **"id"**: **"user000039"**,  **"account"**: **"xman"**,  **"password"**: **"12345"**,  **"email"**: **"spman@gmail.com"**,  **"role"**: {  **"id"**: **"role00003"**,  **"name"**: **"统计用户"** }  }  ],  **"roles"**: [  {  **"id"**: **"role00001"**,  **"menus"**: [  **"/home"**,  **"/login"** ],  **"name"**: **"普通用户"**,  **"state"**: 1  },  {  **"id"**: **"role00002"**,  **"menus"**: [  **"/home"**,  **"/login"** ],  **"name"**: **"vip用户"**,  **"state"**: 1  },  {  **"id"**: **"role00003"**,  **"menus"**: [  **"/home"**,  **"/login"**,  **"/charts"**,  **"charts/bar"**,  **"charts/line"**,  **"charts/pie"** ],  **"name"**: **"统计用户"**,  **"state"**: 1  },  {  **"id"**: **"role00004"**,  **"menus"**: [  **"/home"**,  **"/login"**,  **"/products/list"**,  **"/products/category"** ],  **"name"**: **"普通管理员"**,  **"state"**: 1  },  {  **"id"**: **"role00005"**,  **"menus"**: [  **"/home"**,  **"/login"**,  **"/products"**,  **"/role"**,  **"/products/list"**,  **"products/category"** ],  **"name"**: **"超级管理员"**,  **"state"**: 1  },  {  **"id"**: **"role00006"**,  **"menus"**: [  **"/home"**,  **"/login"**,  **"/products"**,  **"/role"**,  **"/products/list"**,  **"products/category"** ],  **"name"**: **"数据管理员"**,  **"state"**: 1  },  {  **"id"**: **"role00007"**,  **"menus"**: [  **"/home"**,  **"/login"**,  **"/products"**,  **"/role"**,  **"/products/list"**,  **"products/category"** ],  **"name"**: **"网络管理员"**,  **"state"**: 1  }  ],  **"goods"**: [  {  **"id"**: **"goods0000001"**,  **"delState"**: 0,  **"msg"**: **"IPhone 12 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 12 "**,  **"price"**: 999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000002"**,  **"delState"**: 0,  **"msg"**: **"IPhone 13 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 13 "**,  **"price"**: 1999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000003"**,  **"delState"**: 0,  **"msg"**: **"IPhone 13s 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 13s "**,  **"price"**: 21000.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000004"**,  **"delState"**: 0,  **"msg"**: **"IPhone 13c 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 13c "**,  **"price"**: 19999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000005"**,  **"delState"**: 0,  **"msg"**: **"IPhone 14 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 14 "**,  **"price"**: 2200.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000006"**,  **"delState"**: 0,  **"msg"**: **"IPhone 14c 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 14c "**,  **"price"**: 1000.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000007"**,  **"delState"**: 0,  **"msg"**: **"IPhone 15 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 15 "**,  **"price"**: 1500.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000008"**,  **"delState"**: 0,  **"msg"**: **"IPhone 15c 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPhone 15c "**,  **"price"**: 2400.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000009"**,  **"delState"**: 0,  **"msg"**: **"IPad 8 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPad 8 "**,  **"price"**: 380.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000010"**,  **"delState"**: 0,  **"msg"**: **"IPad 9 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPad 8"**,  **"price"**: 390.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000011"**,  **"delState"**: 0,  **"msg"**: **"IPad 10 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"IPad 8 "**,  **"price"**: 400.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000012"**,  **"delState"**: 0,  **"msg"**: **"三星galaxy4手机 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"三星galaxy4手机 "**,  **"price"**: 1100.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000013"**,  **"delState"**: 0,  **"msg"**: **"三星galaxy5手机 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"三星galaxy4手机"**,  **"price"**: 9999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000014"**,  **"delState"**: 0,  **"msg"**: **"华为mate40手机 后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"华为mate40手机"**,  **"price"**: 1999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  },  {  **"id"**: **"goods0000015"**,  **"delState"**: 0,  **"msg"**: **"华为mate50手机，后置双摄像头 6种颜色，宽75.7mm，高159mm，厚20mm"**,  **"name"**: **"华为mate50手机"**,  **"price"**: 2000.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **""** }  }  ] } |

|  |
| --- |
|  |

### 2.在products.js里面添加一个getProductsApi接口

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

### 3.测试一下是否能够获取数据，我们添加一个方法getProducts，然后在useEffect的模拟ComponentDidMount生命周期的代码里面去获取数据，并且输出到控制台

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table*** } **from 'antd'**; **import** {*getProductsApi*} **from "../../../api/products"**;  **const** columns = [  {  **title**: **'Name'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'Age'**,  **dataIndex**: **'age'**,  },  {  **title**: **'Address'**,  **dataIndex**: **'address'**,  }, ]; **const** data = [  {  **key**: **'1'**,  **name**: **'John Brown'**,  **age**: 32,  **address**: **'New York No. 1 Lake Park'**,  },  {  **key**: **'2'**,  **name**: **'Jim Green'**,  **age**: 42,  **address**: **'London No. 1 Lake Park'**,  },  {  **key**: **'3'**,  **name**: **'Joe Black'**,  **age**: 32,  **address**: **'Sydney No. 11 Lake Park'**,  },  {  **key**: **'4'**,  **name**: **'John'**,  **age**: 29,  **address**: **'Sydney No. 1 Lake Park'**,  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  } };  **const** *ProductsList* = () => {  *//状态* **let** [productsData,setProductsData] = *useState*([])  *//方法* **let** getProducts = **async** ()=>{  **let** data = **await** *getProductsApi*()  ***console***.log(data)  }    *useEffect*(()=>{  getProducts()  },[])  **return** (  <**div**>  <**Table  rowSelection=**{{**type**: **'checkbox'**, ...rowSelection}}  **columns=**{columns}  **dataSource=**{data}  />  </**div**>  ); };  **export default** *ProductsList*; |

#### 效果如下，

|  |
| --- |
|  |

#### 说明接口没有问题

## 4）根据我们的数据格式修改表格的数据并且用我们的数据来渲染，注意如果需要使用到嵌套属性需要使用数组的形式

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table***,***Space*** } **from 'antd'**; **import** {*getProductsApi*} **from "../../../api/products"**;  **const** columns = [  {  **title**: **'商品名称'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'描述'**,  **dataIndex**: **'title'**,  },  {  **title**: **'价格'**,  **dataIndex**: **'price'**,  },  {  **title**: **'类别'**,  **dataIndex**: [**'type'**,**'name'**],  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:()=>{  **return** (  <**Space**>  <**a**>修改</**a**>  <**a**>删除</**a**>  </**Space**>  )  }  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  } };  **const** *ProductsList* = () => {  *//状态* **let** [productsData,setProductsData] = *useState*([])  *//方法* **let** getProducts = **async** ()=>{  **let** data = **await** *getProductsApi*()  *// console.log(data)* setProductsData([...data])  ***console***.log(**"products:"**,productsData)  }   *useEffect*(()=>{  getProducts()   },[])  **return** (  <**div**>  <**Table  rowSelection=**{{**type**: **'checkbox'**, ...rowSelection}}  **columns=**{columns}  **dataSource=**{productsData}  **rowKey=**{**"id"**}  />  </**div**>  ); };  **export default** *ProductsList*; |

#### 需要注意的是：商品数据比用户数据多得多，不能够简单的利用一个弹窗就能搞定。需要设置利用进行页面跳转

## 5）我们在products文件夹下面新建一个ProductsUpdate.jsx

|  |
| --- |
|  |

## 6)需要为这个页面配置路由，我们打开App.jsx.在里面配置路由

|  |
| --- |
| **import *React***, {***Suspense***,*lazy*} **from 'react'**; **import** {BrowserRouter,HashRouter,Route,Routes,Navigate} **from "react-router-dom"**; **import** *LoginPage* **from "./pages/login/LoginPage"**; **import** *IndexPage* **from "./pages/index/IndexPage"**; **import** *HomePage* **from "./pages/index/home/HomePage"**; **import** *UsersPage* **from "./pages/index/users/UsersPage"**; **import** *RolesPage* **from "./pages/index/roles/RolesPage"**; **import** *ProductsCategory* **from "./pages/index/products/ProductsCategory"**; **import** *ProductsList* **from "./pages/index/products/ProductsList"**; **import** *ProductsUpdate* **from "./pages/index/products/ProductsUpdate"**;    **let** NotFound = *lazy*(()=>**import**(**"./pages/404/NotFound"**));  **const** *App* = () => {  **return** (  <**div**>  <**Suspense fallback=**{<**h1**>加载中...</**h1**>}>  <**BrowserRouter**>  <**Routes**>  <**Route path="/" element =**{<**Navigate to="/home"**/>}></**Route**>  <**Route path="/" element=**{<**IndexPage**/>}>  <**Route path="home" element=**{<**HomePage**/>}></**Route**>  <**Route path="users" element=**{<**UsersPage**/>}></**Route**>  <**Route path="roles" element=**{<**RolesPage**/>}></**Route**>  <**Route path="products/category" element=**{<**ProductsCategory**/>}></**Route**>  <**Route path="products/list" element=**{<**ProductsList**/>}></**Route**>  <**Route path="products/update" element=**{<**ProductsUpdate**/>}></**Route**>  </**Route**>  <**Route path="/login" element=**{<**LoginPage**/>}></**Route**>  <**Route path='\*' element=**{<**NotFound**/>}></**Route**>  </**Routes**>  </**BrowserRouter**>  </**Suspense**>   </**div**>  ); };  **export default** *App*; |

## 7）回到ProductsList.jsx页面，把修改的a并且改为Link，并且设置to属性为需要跳转的路径

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table***,***Space*** } **from 'antd'**; **import** {*getProductsApi*} **from "../../../api/products"**; **import** {*Link*} **from "react-router-dom"**;  **const** columns = [  {  **title**: **'商品名称'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'描述'**,  **dataIndex**: **'title'**,  },  {  **title**: **'价格'**,  **dataIndex**: **'price'**,  },  {  **title**: **'类别'**,  **dataIndex**: [**'type'**,**'name'**],  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:()=>{  **return** (  <**Space**>  <**Link to="/products/update"**>修改</**Link**>  <**a**>删除</**a**>  </**Space**>  )  }  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  } };  **const** *ProductsList* = () => {  *//状态* **let** [productsData,setProductsData] = *useState*([])  *//方法* **let** getProducts = **async** ()=>{  **let** data = **await** *getProductsApi*()  *// console.log(data)* setProductsData([...data])  ***console***.log(**"products:"**,productsData)  }   *useEffect*(()=>{  getProducts()   },[])  **return** (  <**div**>  <**Table  rowSelection=**{{**type**: **'checkbox'**, ...rowSelection}}  **columns=**{columns}  **dataSource=**{productsData}  **rowKey=**{**"id"**}  />  </**div**>  ); };  **export default** *ProductsList*; |

### 效果：点击修改会跳转到update页面

|  |
| --- |
|  |

#### 现在需要解决的问题就是：update页面需要有表单，而且表单需要回显数据，这就需要路由传参，

## React Router提供了2种比较好用的传参方式：

### 1.动态路由

#### a.给render方法一个形参来接收需要修改的对象的数据，然后利用路由拼接把id拼接到路径中

|  |
| --- |
| **import *React***, {*useEffect*, *useState*} **from 'react'**; **import** { ***Divider***, ***Radio***, ***Table***,***Space*** } **from 'antd'**; **import** {*getProductsApi*} **from "../../../api/products"**; **import** {*Link*} **from "react-router-dom"**;  **const** columns = [  {  **title**: **'商品名称'**,  **dataIndex**: **'name'**,  render: (text) => <**a**>{text}</**a**>,  },  {  **title**: **'描述'**,  **dataIndex**: **'title'**,  },  {  **title**: **'价格'**,  **dataIndex**: **'price'**,  },  {  **title**: **'类别'**,  **dataIndex**: [**'type'**,**'name'**],  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record)=>{  **return** (  <**Space**>  <**Link to=**{**"/products/update/"**+record.**id**}>修改</**Link**>  <**a**>删除</**a**>  </**Space**>  )  }  }, ];  *// rowSelection object indicates the need for row selection* **const** rowSelection = {  onChange: (selectedRowKeys, selectedRows) => {  ***console***.log(**`selectedRowKeys:** ${selectedRowKeys}**`**, **'selectedRows: '**, selectedRows);  } };  **const** *ProductsList* = () => {  *//状态* **let** [productsData,setProductsData] = *useState*([])  *//方法* **let** getProducts = **async** ()=>{  **let** data = **await** *getProductsApi*()  *// console.log(data)* setProductsData([...data])  ***console***.log(**"products:"**,productsData)  }   *useEffect*(()=>{  getProducts()   },[])  **return** (  <**div**>  <**Table  rowSelection=**{{**type**: **'checkbox'**, ...rowSelection}}  **columns=**{columns}  **dataSource=**{productsData}  **rowKey=**{**"id"**}  />  </**div**>  ); };  **export default** *ProductsList*; |

#### 效果，看浏览器路径，此时匹配不上，显示404

|  |
| --- |
|  |

#### b.动态路由的配置，在App.jsx里面把新添加的路由改动一下

|  |
| --- |
| **import *React***, {***Suspense***,*lazy*} **from 'react'**; **import** {BrowserRouter,HashRouter,Route,Routes,Navigate} **from "react-router-dom"**; **import** *LoginPage* **from "./pages/login/LoginPage"**; **import** *IndexPage* **from "./pages/index/IndexPage"**; **import** *HomePage* **from "./pages/index/home/HomePage"**; **import** *UsersPage* **from "./pages/index/users/UsersPage"**; **import** *RolesPage* **from "./pages/index/roles/RolesPage"**; **import** *ProductsCategory* **from "./pages/index/products/ProductsCategory"**; **import** *ProductsList* **from "./pages/index/products/ProductsList"**; **import** *ProductsUpdate* **from "./pages/index/products/ProductsUpdate"**;    **let** NotFound = *lazy*(()=>**import**(**"./pages/404/NotFound"**));  **const** *App* = () => {  **return** (  <**div**>  <**Suspense fallback=**{<**h1**>加载中...</**h1**>}>  <**BrowserRouter**>  <**Routes**>  <**Route path="/" element =**{<**Navigate to="/home"**/>}></**Route**>  <**Route path="/" element=**{<**IndexPage**/>}>  <**Route path="home" element=**{<**HomePage**/>}></**Route**>  <**Route path="users" element=**{<**UsersPage**/>}></**Route**>  <**Route path="roles" element=**{<**RolesPage**/>}></**Route**>  <**Route path="products/category" element=**{<**ProductsCategory**/>}></**Route**>  <**Route path="products/list" element=**{<**ProductsList**/>}></**Route**>  <**Route path="products/update/:id" element=**{<**ProductsUpdate**/>}></**Route**>  </**Route**>  <**Route path="/login" element=**{<**LoginPage**/>}></**Route**>  <**Route path='\*' element=**{<**NotFound**/>}></**Route**>  </**Routes**>  </**BrowserRouter**>  </**Suspense**>   </**div**>  ); };  **export default** *App*; |

#### 效果，此时就正常了

|  |
| --- |
|  |

#### c.接收参数

参数是传递过来了，我们需要接收和处理参数，这么接收这个参数？需要用到react-router-dom的hook：useParam。

|  |
| --- |
| **import *React* from 'react'**; **import** {*useParams*} **from "react-router-dom"**;  **const** *ProductsUpdate* = () => {  *//接收参数* **let** params = *useParams*();  ***console***.log(params)  **return** (  <**div**>  update 页面  </**div**>  ); };  **export default** *ProductsUpdate*; |
|  |

#### 然后根据得到的id从后端获取需要修改的产品信息。我们在products.js中封装一个根据id查询商品的接口，我和老师的不一样，老师的是post方法，我的是get方法

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

#### 我的接口，products.js

|  |
| --- |
| **import *request* from "../utils/request"**;  **export let** *getProductsApi* = ()=>***request***.get(**"/goods"**) **export let** *findProductByIdApi* =(id)=>***request***.get(**`/goods/**${id}**`**) |

### 2.state传参

#### 1）我们把动态路由传参的代码注释了，把Link改为没有动态参数的状态。

|  |
| --- |
|  |

#### 2）给Link的state属性设置一下，改为一个对象

|  |
| --- |
|  |

#### 3）在App.jsx里面把动态路由还原为普通路由

|  |
| --- |
|  |

#### 4）接收参数，需要用到useLocation，

|  |
| --- |
|  |

#### 5）在useEffect里面的调用方式也需要做一些调整

|  |
| --- |
|  |

##### 效果，也是可以拿到数据的

|  |
| --- |
|  |

##### 这两种方法都可以用，原则是：当需要传递多个参数时，使用state传参会好一点，当参数很少就可以使用动态路由

### 修改数据的表单里面需要使用一个级联选择组件Cascader