# 之前的笔记链接

react全家桶学习笔记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：[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：[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： [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](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：[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](react全家桶学习笔记7-路由传参.docx)

react全家桶学习笔记8: [状态机初始化配置](react全家桶学习笔记8-状态机初始配置.docx)

react全家桶学习笔记9: [redux获取分类数据](react全家桶学习笔记9-redux获取分类数据.docx)

# react全家桶学习笔记10：渲染一二级分类数据以及优化，

# 注意：我们是使用json-server来作为模拟后端，data.json作为后端数据库，需要打开一个cmd窗口，先把json-server启动起来

# React-fam-bucketdemo项目，我们前面已经拿到数据了现在来渲染页面

## 1）到ant design网站找一个表格，这是一个有展开功能的表格，他的展开功能是有数据决定的。只要一个列数据里面有children属性，他就可以展开，我们先把表格的列设置成我们需要的格式，然后把我们获取到的一级分类渲染上去。

### src/pages/index/category/ProductsCategory.jsx

|  |
| --- |
| **import *React***, {*useEffect*,*useState*} **from 'react'**; **import** {***useDispatch***, ***useSelector***} **from "react-redux"**; **import** {*getCategoryAsync*} **from "../../../store/category/action"**; **import** { ***Space***, ***Switch***, ***Table*** } **from 'antd'**;  **const** columns = [  {  **title**: **'分类名称'**,  **dataIndex**: **'name'**,  },  {  **title**: **'类别'**,  **dataIndex**: **'type'**,  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record) =>(  <**Space**>  <**a**>修改</**a**>  <**a**>删除</**a**>  </**Space**>  )    }, ];   **const** *ProductsCategory* = () => {  *//状态数据   useEffect*(()=>{  getCategory()  },[])  *//接收一个dispatch函数，用来调用仓库的公共方法* **let** dispatch = ***useDispatch***()  *//先从仓库里面获取数据，看看是否是空数组* **let** categoryData = ***useSelector***((state)=>{  **return** state.**category** })  ***console***.log(**"categoryData:"**,categoryData)   *//发送异步请求获取数据* **let** getCategory = ()=>{  *//调用自己封装的仓库方法,不能直接调用，需要使用dispatch方法来调用* dispatch(*getCategoryAsync*({**parentId**:0}))   }   **return** (  <>  <**Table columns=**{columns} **dataSource=**{categoryData} **rowKey="id"**/>  </>  ); };  **export default** *ProductsCategory*; |

### 效果：

|  |
| --- |
|  |

#### 这里有一个问题，后端的数据没有children，而展开功能需要我们的数据有childrn属性，那怎么办？办法就是给数据添加children属性

## 2）添加给数据children属性的比较好的方法是使用计算属性，也就是useMemo，我们来实现一下

|  |
| --- |
| **import *React***, {*useEffect*, *useMemo*, *useState*} **from 'react'**; **import** {***useDispatch***, ***useSelector***} **from "react-redux"**; **import** {*getCategoryAsync*} **from "../../../store/category/action"**; **import** { ***Space***, ***Switch***, ***Table*** } **from 'antd'**;  **const** columns = [  {  **title**: **'分类名称'**,  **dataIndex**: **'name'**,  },  {  **title**: **'类别'**,  **dataIndex**: **'type'**,  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record) =>(  <**Space**>  <**a**>修改</**a**>  <**a**>删除</**a**>  </**Space**>  )    }, ];   **const** *ProductsCategory* = () => {   *//先从仓库里面获取数据，看看是否是空数组* **let** categoryData = ***useSelector***((state)=>{  **return** state.**category** })  ***console***.log(**"categoryData:"**,categoryData)   *//计算属性用来给数据添加children属性* **let** category = *useMemo*(()=>{  **return** categoryData.map(item=>{  **return** {  ...item ,*//保留旧属性* **children** : [] *//有了这个属性，表格才有+号* }  })  },[categoryData])  *useEffect*(()=>{  getCategory()  },[])  *//接收一个dispatch函数，用来调用仓库的公共方法* **let** dispatch = ***useDispatch***()   *//发送异步请求获取数据* **let** getCategory = ()=>{  *//调用自己封装的仓库方法,不能直接调用，需要使用dispatch方法来调用* dispatch(*getCategoryAsync*({**parentId**:0}))   }   **return** (  <>  <**Table columns=**{columns} **dataSource=**{category} **rowKey="id"**/>  </>  ); };  **export default** *ProductsCategory*; |

### 效果：

|  |
| --- |
|  |

#### 现在+号是有了，那么我们这么实现当我们点击+号，表格会展开并且就能够获取二级分类的数据渲染上去？

## 3）我们需要到ant design网站里面去查看api，发现一个叫做expandable组件有一个onExpand事件，那么我们指明使用呢，我们再查看一下，会发现它是table的一个属性，我们需要在我们的table里面配置这个属性，配置的代码如下

|  |
| --- |
|  |

## 4）实现onExpand处理函数的功能，为他添加两个形参来接收数据，我们先用log在控制台里面测试一下他们是什么数据

|  |
| --- |
|  |

### 效果

|  |
| --- |
|  |

#### 通过对数据的分析，我们可以发现，expanded就是一个Boolean类型的变量，他表示当前是展开(true)还是折叠(false)状态，record就是当前选中的行的数据

## 5）我们以前有一个小细节，把reducers.js写成了reducer.js,现在改进了，还有我们的数据需要格式又要改了，我们改为下面的数据

### /public/data.json

|  |
| --- |
| {  **"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"**: **"type00001"**,  **"name"**: **"我最爱"**,  **"type"**: **"一级分类"** }  },  {  **"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"**: **"type00002"**,  **"name"**: **"你最爱"**,  **"type"**: **"二级分类"** }  },  {  **"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"**: 2100.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **"type00003"**,  **"name"**: **"他最爱"**,  **"type"**: **"三级分类"** }  },  {  **"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"**: 1999.0,  **"state"**: 1,  **"title"**: **"搭载最新处理器。双面玻璃材质"**,  **"type"**: {  **"id"**: **"type00004"**,  **"name"**: **"真漂亮"**,  **"type"**: **"二级分类"** }  },  {  **"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"**: **"type00005"**,  **"name"**: **"爱不释手"**,  **"type"**: **"一级分类"** }  },  {  **"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"**: **"type00005"**,  **"name"**: **"爱不释手"**,  **"type"**: **"一级分类"** }  },  {  **"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"**: **"type00004"**,  **"name"**: **"真漂亮"**,  **"type"**: **"二级分类"** }  },  {  **"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"**: **"type00004"**,  **"name"**: **"真漂亮"**,  **"type"**: **"二级分类"** }  },  {  **"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"**: **"type00006"**,  **"name"**: **"主推品"**,  **"type"**: **"一级分类"** }  },  {  **"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"**: **"type00004"**,  **"name"**: **"真漂亮"**,  **"type"**: **"二级分类"** }  },  {  **"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"**: **"type00007"**,  **"name"**: **"好品质"**,  **"type"**: **"三级分类"** }  },  {  **"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"**: **"type00003"**,  **"name"**: **"他最爱"**,  **"type"**: **"三级分类"** }  },  {  **"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"**: **"type00003"**,  **"name"**: **"他最爱"**,  **"type"**: **"三级分类"** }  },  {  **"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"**: **"type00003"**,  **"name"**: **"他最爱"**,  **"type"**: **"三级分类"** }  },  {  **"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"**: **"type00001"**,  **"name"**: **"我最爱"**,  **"type"**: **"一级分类"** }  }  ],  **"category"**: [  {  **"id"**: **"cat00001"**,  **"name"**: **"珠宝首饰"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },{  **"id"**: **"cat00001-1"**,  **"name"**: **"戒指"**,  **"parentId"**:**"cat00001"**,  **"type"**: **"一级分类"** },   {  **"id"**: **"cat00002"**,  **"name"**: **"数码产品"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00002-1"**,  **"name"**: **"电脑"**,  **"parentId"**: **"cat00002"**,  **"type"**: **"二级分类"** },   {  **"id"**: **"cat00003"**,  **"name"**: **"家用电器"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00003-1"**,  **"name"**: **"电视机"**,  **"parentId"**: **"cat00003"**,  **"type"**: **"二级分类"** },  {  **"id"**: **"cat00004"**,  **"name"**: **"汽车用品"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00004-1"**,  **"name"**: **"电瓶"**,  **"parentId"**: **"cat00004"**,  **"type"**: **"二级分类"** },   {  **"id"**: **"cat00005"**,  **"name"**: **"运动用品"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00005-1"**,  **"name"**: **"运动鞋"**,  **"parentId"**: **"cat00005"**,  **"type"**: **"二级级分类"** },  {  **"id"**: **"cat00006"**,  **"name"**: **"办公用品"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00006-1"**,  **"name"**: **"计算器"**,  **"parentId"**: **"cat00006"**,  **"type"**: **"二级分类"** },  {  **"id"**: **"cat00007"**,  **"name"**: **"休闲用品"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00007"**,  **"name"**: **"休闲鞋"**,  **"parentId"**: **"cat00007"**,  **"type"**: **"二级分类"** },  {  **"id"**: **"cat00008"**,  **"name"**: **"文物周刊"**,  **"parentId"**: 0,  **"type"**: **"一级分类"** },  {  **"id"**: **"cat00008-1"**,  **"name"**: **"小说"**,  **"parentId"**: **"cat00008"**,  **"type"**: **"二级级分类"** }   ],  **"category\_bak"**: [   {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"sj000001"**,  **"value"**: **"手机"**,  **"label"**: **"手机"** }  ]  },  {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"pb000001"**,  **"value"**: **"平板"**,  **"label"**: **"平板"** }  ]  },  {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"sj000002"**,  **"value"**: **"手机充电器"**,  **"label"**: **"手机充电器"** }  ]  },  {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"平板000002"**,  **"value"**: **"平板充电器"**,  **"label"**: **"平板充电器"** }  ]  },  {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"sj000003"**,  **"value"**: **"手机耳机"**,  **"label"**: **"手机二级"** }  ]  },  {  **"value"**: **"数码产品"**,  **"label"**: **"数码产品"**,  **"children"**: [  {  **"id"**: **"pb000003"**,  **"value"**: **"平板耳机"**,  **"label"**: **"平板耳机"** }  ]  }  ] } |

## 6）然后我们在+号的点击处理事件onExpand添加发送请求代码，需要做一些修改，不能直接调用仓库方法，因为直接调用就会造成二级分类数据覆盖一级分类数据的问题。我们需要把二级分类数据添加到一级分类数据下面。我们需要在一级分类的基础上添加二级分类，然后再保存到仓库，这里需要直接调用我们封装的api，然后看看能否拿到数据

|  |
| --- |
|  |

### 效果：

|  |
| --- |
|  |

### 想一想，拿到数据后，怎么处理

## 7）这里需要使用仓库的另外一种用法，因为这里用仓库的异步方法无法完成，我们打开store/category/reducers.js,给他添加处理分支，也就是组件直接调用api然后让

|  |
| --- |
| **export let** *categoryReducer* = (state= [],action) =>{  **switch** (action.**type**) {  **case "SET\_CATEGORY"**:  **return** action.**payload** *//添加二级分类* **case "SET\_SUB\_CATEGORY"**:  *//需要先保留一级分类的数据，然后添加二级分类的数据* **return** state.map(item=>{  **if**(item.**id** === action.**payload**.**parentId**){  **return** {  ...item,  **children**:action.**payload**.**children** }  }  **return** item  })  **default**:  **return** state  }  } |

## 8）然后我们需要在组件里面跳过action，直接把数据给reducer

|  |
| --- |
|  |

### 注意：此时我们在控制台点击任意一个+号发现控制台对应的输出的项有children，但是表格没有扩展，这是为什么，是计算属性的问题，在里面把childen设置成空的

## 9）解决办法1.在计算属性中需要判断每一项是否有children，如果有就不用为他添加children，如果没有就给他添加一个空数组作为children

|  |
| --- |
|  |

#### 效果：现在的效果就正常了

|  |
| --- |
|  |

## ProductsCategory.jsx的代码如下

|  |
| --- |
| **import *React***, {*useEffect*, *useMemo*, *useState*} **from 'react'**; **import** {***useDispatch***, ***useSelector***} **from "react-redux"**; **import** {*getCategoryAsync*, *setSubCategoryAction*} **from "../../../store/category/actions"**; **import** { ***Space***,***Tag***, ***Switch***, ***Table*** } **from 'antd'**; **import** {*getCategoryApi*} **from "../../../api/category"**; **import** {*Link*} **from "react-router-dom"**;   **const** columns = [  {  **title**: **'分类名称'**,  **dataIndex**: **'name'**,  },  {  **title**: **'类别'**,  **dataIndex**: **'type'**,  render:(text)=>{  **if**(text==**"一级分类"**){  **return** <**Tag color="magenta"**>{text}</**Tag**>  }**else if**(text==**"二级分类"**){  **return** <**Tag color="blue"**>{text}</**Tag**>  }  **return** text  }  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record) =>(  <**Space**>  {*/\* 方式一 动态路由\*/*}  {*/\*<Link to={"/products/update/"+ record.id}>修改</Link>\*/*}  {*/\* 方式2 state传参\*/*}  <**Link to="/products/update" state =**{{**id**:record.**id**}}>修改</**Link**>  <**a**>删除</**a**>  </**Space**>  )    }, ];   **const** *ProductsCategory* = () => {   *//先从仓库里面获取数据，看看是否是空数组* **let** categoryData = ***useSelector***((state)=>{  **return** state.**category** })  ***console***.log(**"categoryData:"**,categoryData)   *//显示+号的方式1，用计算属性来给数据添加children属性  // let category = useMemo(()=>{  //  // return categoryData.map(item=>{  // //如果没有children就给他添加一个空数组作为children  // if(!item.children){  // return {  // ...item ,//保留旧属性  // children : [] //有了这个属性，表格才有+号  // }  // }  // //如果有children就直接返回  // return item  // })  // },[categoryData])   //显示+号的方式2 不使用计算属性，需要修改reducers.js的代码，可以在reducers.js中修改代码   useEffect*(()=>{  getCategory()  },[])  *//接收一个dispatch函数，用来调用仓库的公共方法* **let** dispatch = ***useDispatch***()   *//发送异步请求获取数据* **let** getCategory = ()=>{  *//调用自己封装的仓库方法,不能直接调用，需要使用dispatch方法来调用* dispatch(*getCategoryAsync*({**parentId**:0}))   }  *//点击表格列上面的+号的事件处理* **let** onExpand = **async** (expanded,record) => {  ***console***.log(expanded,record)  *//需要根据expanded来判断是否需要发送请求获取二级数据* **if**(expanded){  *//获取数据,这里不能使用仓库的方法，因为它是用来获取一级分类数据的  //这里需要直接调用api* **let** res = **await** *getCategoryApi*({**parentId**:record.**id**})  **if**(res){  ***console***.log(res)  *//用请求结果修改仓库数据，还是需要使用dispatch方法  // dispatch({type:'SET\_SUB\_CATEGORY',payload:{parentId:record.id,children:res}})* dispatch(*setSubCategoryAction*({**parentId**:record.**id**,**children**:res}))   }   }  }   **return** (  <>  <**Table columns=**{columns} **dataSource=**{categoryData} **rowKey="id" expandable=**{{ onExpand }}/>  </>  ); };  **export default** *ProductsCategory*; |

### store/category/actions.js

|  |
| --- |
| **import** {*getCategoryApi*} **from "../../api/category"**;  **export let** *getCategoryAsync* = (params) =>{   **return async** (dispatch)=>{  *//调用我们封装的发送请求获取数据的api，直接调用，需要使用传入的dispatch方法来保存数据  // let res = await getCategoryApi()* **let** res = **await** *getCategoryApi*(params)  *// console.log("分类数据",res)  //保存数据到仓库* dispatch({**type**:**"SET\_CATEGORY"**,**payload**:res})  } } |

### store/category/reducers.js

|  |
| --- |
| **export let** *categoryReducer* = (state= [],action) =>{  **switch** (action.**type**) {  **case "SET\_CATEGORY"**:  **return** action.**payload** *//添加二级分类* **case "SET\_SUB\_CATEGORY"**:  *//需要先保留一级分类的数据，然后添加二级分类的数据* **return** state.map(item=>{  **if**(item.**id** === action.**payload**.**parentId**){  **return** {  ...item,  **children**:action.**payload**.**children** }  }  **return** item  })  **default**:  **return** state  }  } |

## 10）还有一种方法可以代替计算属性，我们把计算属性的代码注释了，然后把store/category/reducers.js的代码修改一下，一开始就给每一条数据添加一个空数组作为初始值

|  |
| --- |
| **export let** *categoryReducer* = (state= [],action) =>{  **switch** (action.**type**) {  **case "SET\_CATEGORY"**:  **return** action.**payload**.map(item=>{ //一开始就给他添加一个空数组作为children  **return** {  ...item,  **children**:[]  }  })  *//添加二级分类* **case "SET\_SUB\_CATEGORY"**:  *//需要先保留一级分类的数据，然后添加二级分类的数据* **return** state.map(item=>{  **if**(item.**id** === action.**payload**.**parentId**){  **return** {  ...item,  **children**:action.**payload**.**children** }  }  **return** item  })  **default**:  **return** state  }  } |

### 注意：此时table的数据源就要改为categoryData了

|  |
| --- |
|  |

## 改进后的ProductsCategory.jsx

|  |
| --- |
| **import *React***, {*useEffect*, *useMemo*, *useState*} **from 'react'**; **import** {***useDispatch***, ***useSelector***} **from "react-redux"**; **import** {*getCategoryAsync*, *setSubCategoryAction*} **from "../../../store/category/actions"**; **import** { ***Space***,***Tag***, ***Switch***, ***Table*** } **from 'antd'**; **import** {*getCategoryApi*} **from "../../../api/category"**; **import** {*Link*} **from "react-router-dom"**;   **const** columns = [  {  **title**: **'分类名称'**,  **dataIndex**: **'name'**,  },  {  **title**: **'类别'**,  **dataIndex**: **'type'**,  render:(text)=>{  **if**(text==**"一级分类"**){  **return** <**Tag color="magenta"**>{text}</**Tag**>  }**else if**(text==**"二级分类"**){  **return** <**Tag color="blue"**>{text}</**Tag**>  }  **return** text  }  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record) =>(  <**Space**>  {*/\* 方式一 动态路由\*/*}  {*/\*<Link to={"/products/update/"+ record.id}>修改</Link>\*/*}  {*/\* 方式2 state传参\*/*}  <**Link to="/products/update" state =**{{**id**:record.**id**}}>修改</**Link**>  <**a**>删除</**a**>  </**Space**>  )    }, ];   **const** *ProductsCategory* = () => {   *//先从仓库里面获取数据，看看是否是空数组* **let** categoryData = ***useSelector***((state)=>{  **return** state.**category** })  ***console***.log(**"categoryData:"**,categoryData)   *//显示+号的方式1，用计算属性来给数据添加children属性  // let category = useMemo(()=>{  //  // return categoryData.map(item=>{  // //如果没有children就给他添加一个空数组作为children  // if(!item.children){  // return {  // ...item ,//保留旧属性  // children : [] //有了这个属性，表格才有+号  // }  // }  // //如果有children就直接返回  // return item  // })  // },[categoryData])   //显示+号的方式2 不使用计算属性，需要修改reducers.js的代码，可以在reducers.js中修改代码   useEffect*(()=>{  getCategory()  },[])  *//接收一个dispatch函数，用来调用仓库的公共方法* **let** dispatch = ***useDispatch***()   *//发送异步请求获取数据* **let** getCategory = ()=>{  *//调用自己封装的仓库方法,不能直接调用，需要使用dispatch方法来调用* dispatch(*getCategoryAsync*({**parentId**:0}))   }  *//点击表格列上面的+号的事件处理* **let** onExpand = **async** (expanded,record) => {  ***console***.log(expanded,record)  *//需要根据expanded来判断是否需要发送请求获取二级数据* **if**(expanded){  *//获取数据,这里不能使用仓库的方法，因为它是用来获取一级分类数据的  //这里需要直接调用api* **let** res = **await** *getCategoryApi*({**parentId**:record.**id**})  **if**(res){  ***console***.log(res)  *//用请求结果修改仓库数据，还是需要使用dispatch方法  // dispatch({type:'SET\_SUB\_CATEGORY',payload:{parentId:record.id,children:res}})* dispatch(*setSubCategoryAction*({**parentId**:record.**id**,**children**:res}))   }   }  }   **return** (  <>  <**Table columns=**{columns} **dataSource=**{categoryData} **rowKey="id" expandable=**{{ onExpand }}/>  </>  ); };  **export default** *ProductsCategory*; |

# 优化处理

## 1.数据是渲染出来了，但是我们想把一级分类和二级分类做一下区别，渲染不同颜色，我们在ProductsCategory.jsx里面找到类别这一列，添加一个render方法，根据类别来设置样式，需要用到antd的Tag组件

|  |
| --- |
|  |

## 2.状态机可以优化，我们查看一下我们的代码，会发现只要是操作仓库的数据，都是用dispatch方法里面传递一个action对象。所以我们可以把这个action对象封装一下，方便使用。需要在actions.js文件里面写一个函数来返回一个action,然后把组件里面的action参数也封装成一个方法写入actions.js

## store/category/actions.js

|  |
| --- |
| **import** {*getCategoryApi*} **from "../../api/category"**;  *//设置action的方法* **let** setCategoryAction = (data) =>{  **return** {  **type**:**"SET\_CATEGORY"**,  **payload**:data  } } *//将组件里面的actioncs也封装成一个方法* **let** setSubCategoryAction = (data) =>{  **return** {  **type**:**"SET\_SUB\_CATEGORY"**,  **payload**:data  } }  **export let** *getCategoryAsync* = (params) =>{   **return async** (dispatch)=>{  *//调用我们封装的发送请求获取数据的api，直接调用，需要使用传入的dispatch方法来保存数据  // let res = await getCategoryApi()* **let** res = **await** *getCategoryApi*(params)  *// console.log("分类数据",res)  //保存数据到仓库* dispatch(setCategoryAction(res))  } } |

## 3.需要适当修改一下组件的代码

### src/pages/index/products/ProductsCategory.jsx

|  |
| --- |
| **import *React***, {*useEffect*, *useMemo*, *useState*} **from 'react'**; **import** {***useDispatch***, ***useSelector***} **from "react-redux"**; **import** {*getCategoryAsync*, *setSubCategoryAction*} **from "../../../store/category/actions"**; **import** { ***Space***,***Tag***, ***Switch***, ***Table*** } **from 'antd'**; **import** {*getCategoryApi*} **from "../../../api/category"**; **import** {*Link*} **from "react-router-dom"**;   **const** columns = [  {  **title**: **'分类名称'**,  **dataIndex**: **'name'**,  },  {  **title**: **'类别'**,  **dataIndex**: **'type'**,  render:(text)=>{  **if**(text==**"一级分类"**){  **return** <**Tag color="magenta"**>{text}</**Tag**>  }**else if**(text==**"二级分类"**){  **return** <**Tag color="blue"**>{text}</**Tag**>  }  **return** text  }  },  {  **title**: **'操作'**,  **dataIndex**: **'action'**,  render:(\_,record) =>(  <**Space**>  {*/\* 方式一 动态路由\*/*}  {*/\*<Link to={"/products/update/"+ record.id}>修改</Link>\*/*}  {*/\* 方式2 state传参\*/*}  <**Link to="/products/update" state =**{{**id**:record.**id**}}>修改</**Link**>  <**a**>删除</**a**>  </**Space**>  )    }, ];   **const** *ProductsCategory* = () => {   *//先从仓库里面获取数据，看看是否是空数组* **let** categoryData = ***useSelector***((state)=>{  **return** state.**category** })  ***console***.log(**"categoryData:"**,categoryData)   *//显示+号的方式1，用计算属性来给数据添加children属性  // let category = useMemo(()=>{  //  // return categoryData.map(item=>{  // //如果没有children就给他添加一个空数组作为children  // if(!item.children){  // return {  // ...item ,//保留旧属性  // children : [] //有了这个属性，表格才有+号  // }  // }  // //如果有children就直接返回  // return item  // })  // },[categoryData])   //显示+号的方式2 不使用计算属性，需要修改reducers.js的代码，可以在reducers.js中修改代码   useEffect*(()=>{  getCategory()  },[])  *//接收一个dispatch函数，用来调用仓库的公共方法* **let** dispatch = ***useDispatch***()   *//发送异步请求获取数据* **let** getCategory = ()=>{  *//调用自己封装的仓库方法,不能直接调用，需要使用dispatch方法来调用* dispatch(*getCategoryAsync*({**parentId**:0}))   }  *//点击表格列上面的+号的事件处理* **let** onExpand = **async** (expanded,record) => {  ***console***.log(expanded,record)  *//需要根据expanded来判断是否需要发送请求获取二级数据* **if**(expanded){  *//获取数据,这里不能使用仓库的方法，因为它是用来获取一级分类数据的  //这里需要直接调用api* **let** res = **await** *getCategoryApi*({**parentId**:record.**id**})  **if**(res){  ***console***.log(res)  *//用请求结果修改仓库数据，还是需要使用dispatch方法  // dispatch({type:'SET\_SUB\_CATEGORY',payload:{parentId:record.id,children:res}})* dispatch(*setSubCategoryAction*({**parentId**:record.**id**,**children**:res}))   }   }  }   **return** (  <>  <**Table columns=**{columns} **dataSource=**{categoryData} **rowKey="id" expandable=**{{ onExpand }}/>  </>  ); };  **export default** *ProductsCategory*; |

#### 在实际开发中，凡是需要用到的action对象都需要放到actions.js文件中封装成一个方法，这样做是为了统一接口防止调用混乱。

## 4.其实我们看看action里面的两个set方法和reduces.js的方法，发现action的方法的第一个参数的类型需要和reduces.js的case的类型对应，如果我们在每一个的地方都需要自己写，可能会出现少写一个字母而引起不一致的错误，所以我们可以type参数提取出来，我们在store/category下面新建一个actionTypes.js文件，在里面定义所有的操作类型,

|  |
| --- |
| **export const *SET\_CATEGORY*** = **"SET\_CATEGORY" export const *SET\_SUB\_CATEGORY*** = **"SET\_SUB\_CATEGORY"** |

## 5.然后在所有需要使用type的地方用常量来代替

### store/category/actions.js

|  |
| --- |
| **import** {*getCategoryApi*} **from "../../api/category"**; **import** {***SET\_CATEGORY***, ***SET\_SUB\_CATEGORY***} **from "./actionTypes"**;  *//设置action的方法* **let** setCategoryAction = (data) =>{  **return** {  **type**:***SET\_CATEGORY***,  **payload**:data  } } *//将组件里面的actioncs也封装成一个方法* **export let** *setSubCategoryAction* = (data) =>{  **return** {  **type**:***SET\_SUB\_CATEGORY***,  **payload**:data  } }  **export let** *getCategoryAsync* = (params) =>{   **return async** (dispatch)=>{  *//调用我们封装的发送请求获取数据的api，直接调用，需要使用传入的dispatch方法来保存数据  // let res = await getCategoryApi()* **let** res = **await** *getCategoryApi*(params)  *// console.log("分类数据",res)  //保存数据到仓库* dispatch(setCategoryAction(res))  } } |

### store/category/reducers.js

|  |
| --- |
| **import** {***SET\_CATEGORY***, ***SET\_SUB\_CATEGORY***} **from "./actionTypes"**;  **export let** *categoryReducer* = (state= [],action) =>{  **switch** (action.**type**) {  **case *SET\_CATEGORY***:  **return** action.**payload**.map(item=>{  **return** {  ...item,  **children**:[]  }  })  *//添加二级分类* **case *SET\_SUB\_CATEGORY***:  *//需要先保留一级分类的数据，然后添加二级分类的数据* **return** state.map(item=>{  **if**(item.**id** === action.**payload**.**parentId**){  **return** {  ...item,  **children**:action.**payload**.**children** }  }  **return** item  })  **default**:  **return** state  }  } |

### 这样做就比较保险，一次编写，多次调用而且不会出现少敲字母的错误