React,axios,redux,React-Redux，React-router,react-hooks，next.js,



打包：

// 向外暴露一个打包的配置对象； 因为 webpack 是基于Node构建的；所以 webpack 支持所有Node API 和语法

// webpack 默认只能打包处理 .js 后缀名类型的文件； 像 .png .vue 无法主动处理，所以要配置第三方的loader；

// webpack-dev-server 打包好的 main.js 是托管到了内存中；所以在项目根目录中看不到；

// 但是，我们可以认为，在项目根目录中，有一个看不见的 main.js

<!-- <script src="../dist/main.js"></script> -->

<!-- <script src="/main.js"></script> -->

React有bom和dom

// 2. 创建虚拟DOM元素

// 参数1： 创建的元素的类型，字符串，表示元素的名称

// 参数2： 是一个对象或 null, 表示 当前这个 DOM 元素的属性

// 参数3： 子节点（包括 其它 虚拟DOM 获取 文本子节点）

// 参数n: 其它子节点

// <h1 id="myh1" title="this is a h1">这是一个大大的H1</h1>

// const myh1 = React.createElement('h1', null, '这是一个大大的H1')

const myh1 = React.createElement('h1', { id: 'myh1', title: 'this is a h1' }, '这是一个大大的H1')

const mydiv = React.createElement('div', null, '这是一个div元素', myh1)

出现问题：麻烦--我们可以直接在js里面写html代码

const mytest = <div>aaa</div>=====>jsx语法

/ 3. 使用 ReactDOM 把虚拟DOM 渲染到页面上

// 参数1： 要渲染的那个虚拟DOM元素

// 参数2： 指定页面上的DOM元素，当作容器

ReactDOM.render(mytest, document.getElementById('app'))

当我们需要在 JSX 控制的区域内，写 JS 表达式了，则需要把 JS 代码写到 {} 中

  // componentWillMount和componentDidMount这两个生命周期函数，只在页面刷新时执行一次render函数是只要有state和props变化就会执行

  // 0- componentWillReceiveProps(){---如果本组件是一个底层子组件接受props 第一次存在他不会执行，但是当他的dom更新才会执行这个函数

  // 1-shouldComponentUpdate---组件发生改变前执行，可以优化组件性能 利用参数：nextProps:变化后的属性;nextState:变化后的状态;

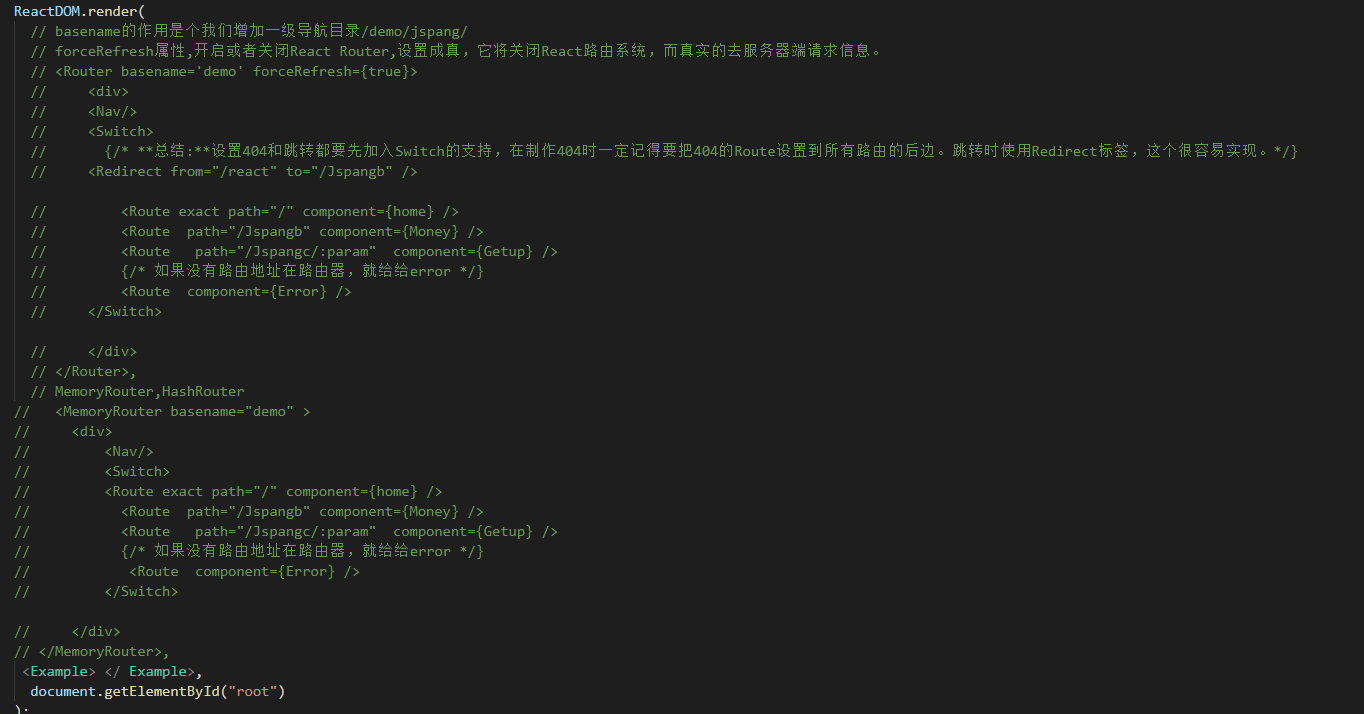
  // 2-componentWillUpdate---组件更新前，shouldComponentUpdate函数之后执行

  // 3-render----开始挂载渲染

  // 4-componentDidUpdate----组件更新之后执行

//componentWillUnmount当dom从页面中删除的时候执行

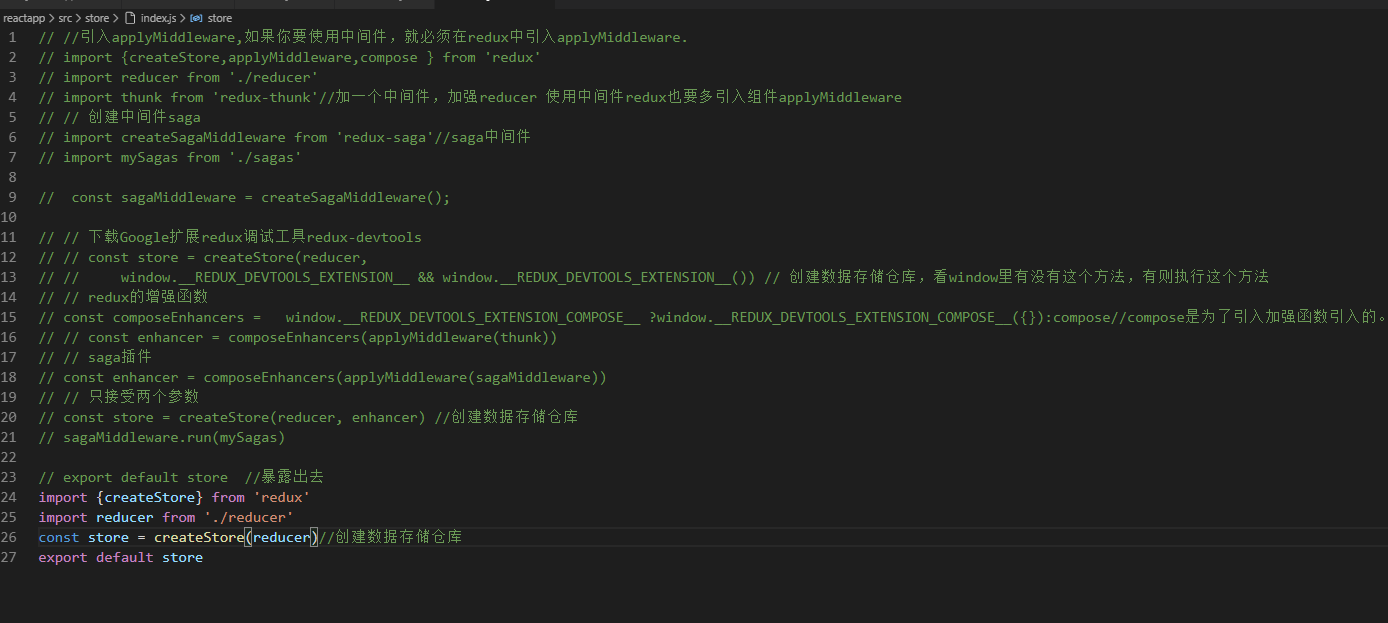
Router



Redux 数据统一管理结合UI界面Ant Design 是react的UI 框架

 zgl 123456'https://www.fastmock.site/mock/759b4f73684cd7a5b5eb36794a121be8/fastmock/api/first/login'

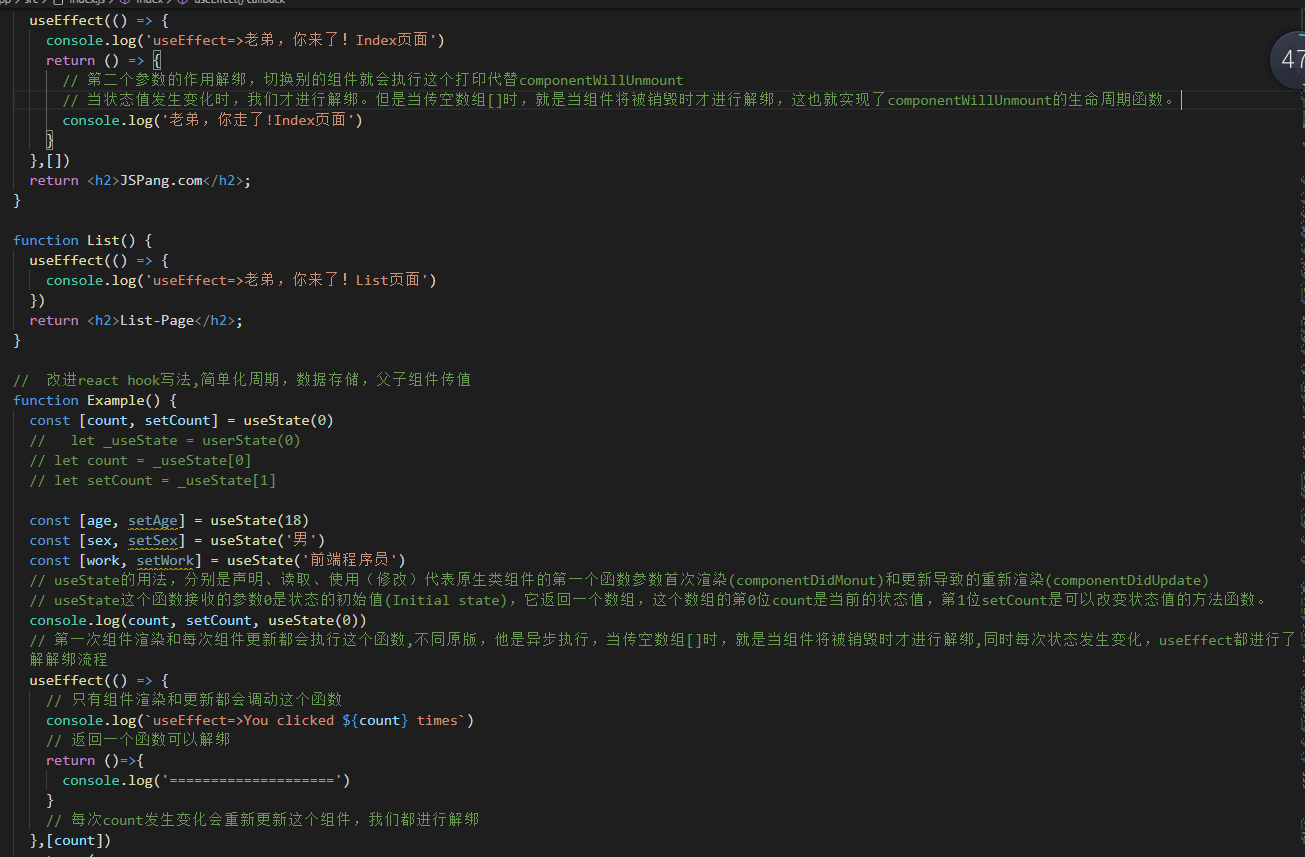
Es6 generration函数

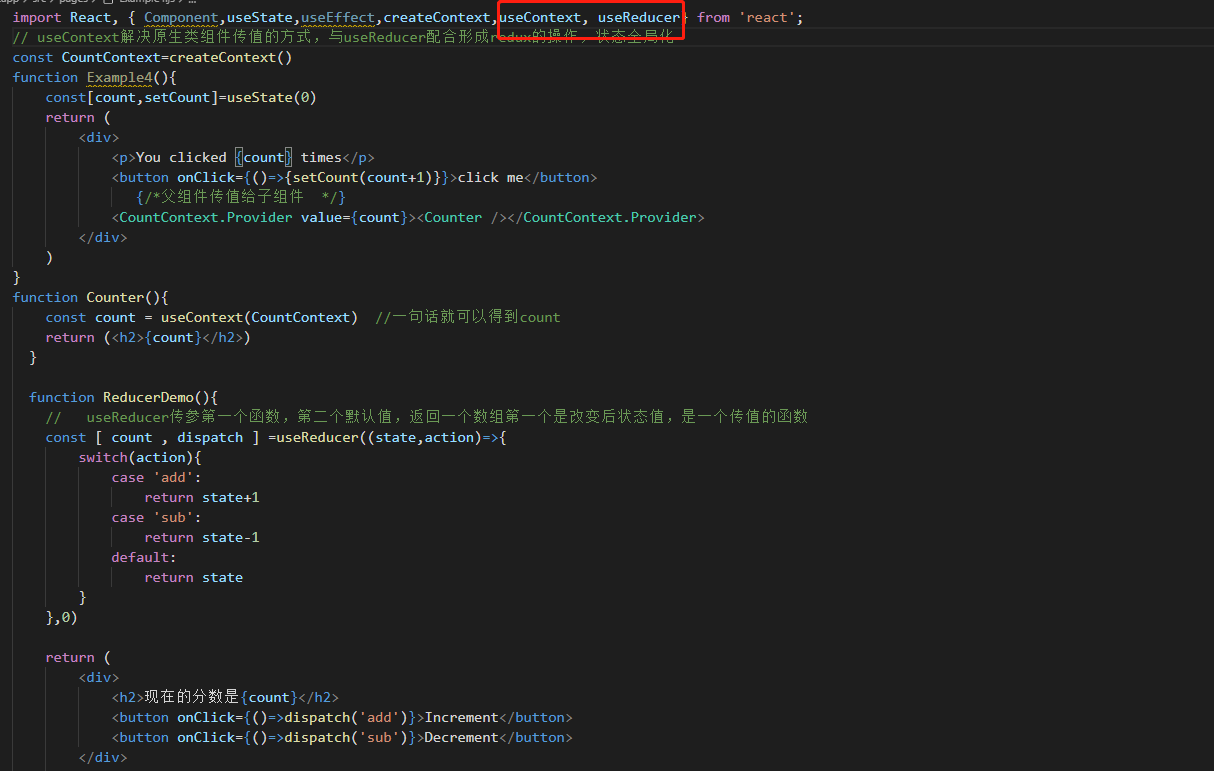


Rudux 中间件 redux-thunk,redux-sagas ,react-redux【】



React-hook







Next.js koas[egg.js]

Next.js.基于服务端的渲染框架。

create-next-app搭建环境

配置环境，他不可以加载css等静态文件，按需加载文件配置

Koa.基于node的服务端。Edd.js基于koa的的上层框架【上层框架就是在Koa的基础上，封装的框架】

npm i egg-init -g

npm install

npm run dev

Egg.js框架就是在Koa2的基础上规范了这些约定，所以也带来了一些文件目录的限制

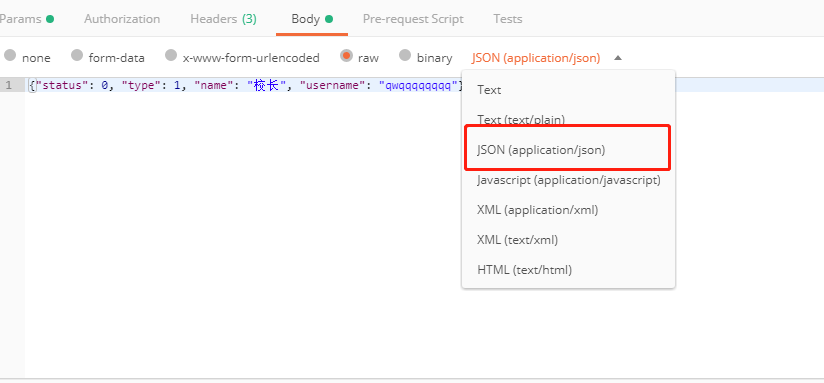
数据库phpstudy服务器

Post请求

数据放在body请求如果是json 需要把数据转化为json格式在传过去

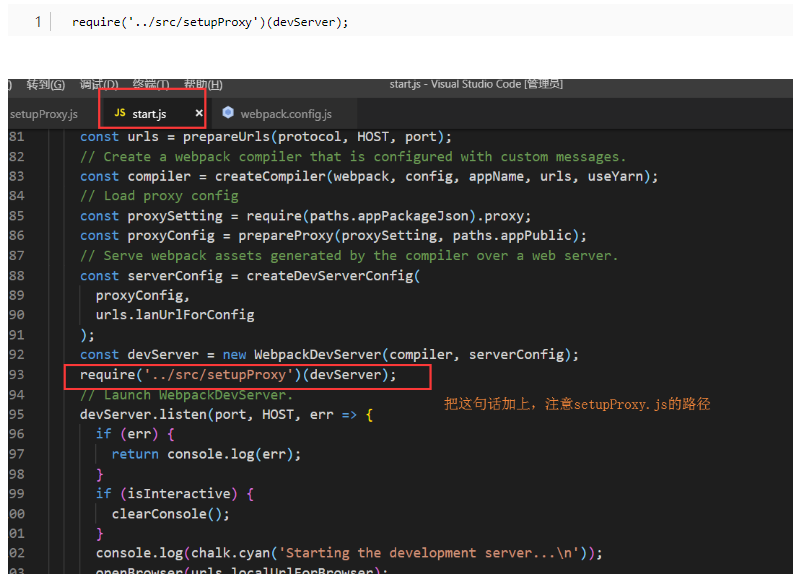
如果

数据如果不放在body放在URL 以params拼接在URL传过去。两种方式不一样



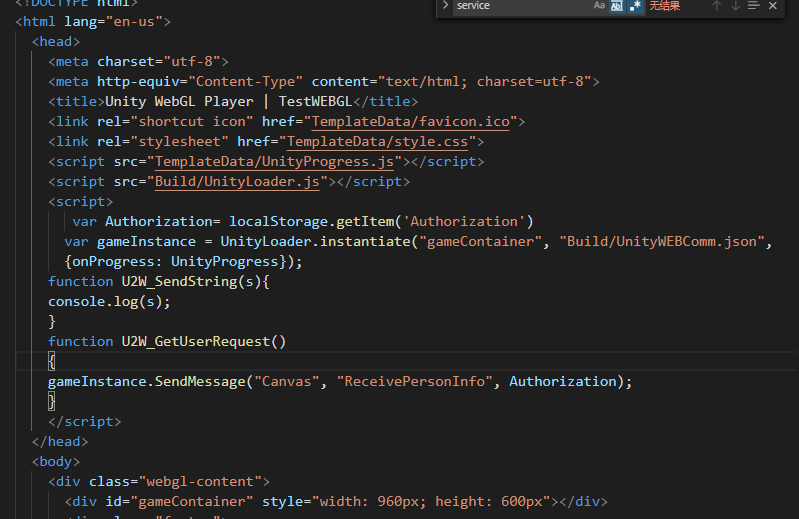
React 解决create-react - app 解决跨域





在service.js 引入这个文件

Webgl和HTML通信



React-create-app 使用webgl用法 【研究插件发送send发送消息的方法不成功】

