前端设置:

1. 安装react-router, socket.io-client, react-scroll-to-bottom, react-emoji, query-string
2. 分别建立join和chat组件(默认路由是join, /chat是chat的路由)

后端设置:

1. 安装cors, nodemon, express, socket.io
2. Require express, http, socketio
3. 用http连接server, 并监听端口
4. 建立路由，并把路由作为middleware使用
5. 拿到io, 用io.on这个函数, 传入socket进行connection, 同时写出disconnection 的情况 (当用户进入房间，connection event执行。当用户离开房间，disconnection event执行)

Join页面前端:

1. 用react钩子定义用户名和房间名
2. 点击进入按钮url会到/chat?用户名&房间名

Chat 页面前端:

* 当页面载入后(useEffect)，用querystring拿到当前页面url的name和room (location.search)
* 拿到socket, 并把name和room emit到后端, 到后端.on接收
* 回到前端的useEffect中，return里把socket disconnect(记住要给useEffect加条件，只有当location.search或endpoint更新的时候才更新)
* useEffect执行结束， 中断socket连接并关闭

后端增删改查:

* 建立添加用户(需要id, name, room)，删除用户(需要id)，获取用户(需要id)，获取当前房间的用户(需要当前房间) 这几个功能性函数
* 从添加用户的返回值中拿到这个用户和可能有的错误， 如果有错的处理此错误
* 否则，用.join()方法把用户添加到房间中

\*\*\*用户发送信息:

后端：

* 定义一个emit为message, 分别用普通方法和boardcast方法把用户和显示信息传到前端, 普通是用户自己看到的显示信息，boardcast是除了用户之外的所有人看到的显示信息

前端:

* 去到前端，设置message和messageList(存放单个message的数组)的useState, 新定义一个useEffect接收message, 把message添加到messageList数组里
* 在jsx中建个输入框，把输入框的内容设置为message，并且回车提交输入框触发sendMessage()函数
* sendMessage()中定义一个叫sendMessage的emit， 把当前message和清空当前输入框的callback传到后端

后端:

* 接收sendMessage emit, 拿到当前用户
* 最后再建一个叫message的emit，把拿到的用户名和用户发的message一起传到整个服务器

前端样式渲染:

* 建立一个infoBar组件，位于对话框上方来显示用户在线状态以及房间名
* 把chat中的input封装成一个表单组件, 再加个和回车功能一样的提交按钮
* 建立对话内容区组件:

把messageList组件化，并循环渲染每一条message

把每一条message再组件化，传入message和 当前的name

如果用户是当前这个name(代表是你在输入), 你输入的信息就会显示到右侧，如果不是当前用户(和你聊天的人), 她的输入信息会显示在左侧

后端移除用户: