Content

[1. Debug reactjs component life cycle 1](#_Toc26564)

[Step 1: add scripts to package.json 2](#_Toc14059)

[Step2: config launch.json 2](#_Toc18568)

[Step3: npm start 3](#_Toc7466)

[Step 4: put break points on source/component/login 3](#_Toc11972)

[Step 5：click 2 launch in step 2 3](#_Toc1955)

[Step 6 start debugger 3](#_Toc24435)

[2. Transform html to Reactjs 4](#_Toc20079)

[Step1: init project 4](#_Toc1316)

[Step2 convert all ejs template to class components 5](#_Toc24889)

[Step 3: add css and js to public folder 8](#_Toc27341)

[Step 4: replace all class= to className= 8](#_Toc22861)

[Step 5 remove all style 9](#_Toc32264)

[3. Add router 9](#_Toc18768)

[Step1: add route wrapper to index.js 9](#_Toc23677)

[Step 2: add route switch to app.js 10](#_Toc8091)

[Step 3 implement hyperlink in react route 10](#_Toc3391)

[4. Add Nodejs login to React 11](#_Toc22670)

[Step 1: how to implement the form 11](#_Toc20065)

[Step2 form handler login - post request 13](#_Toc26709)

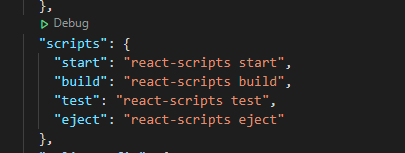
[Step 3 add route context 13](#_Toc24610)

[Step 4 add auth 14](#_Toc16370)

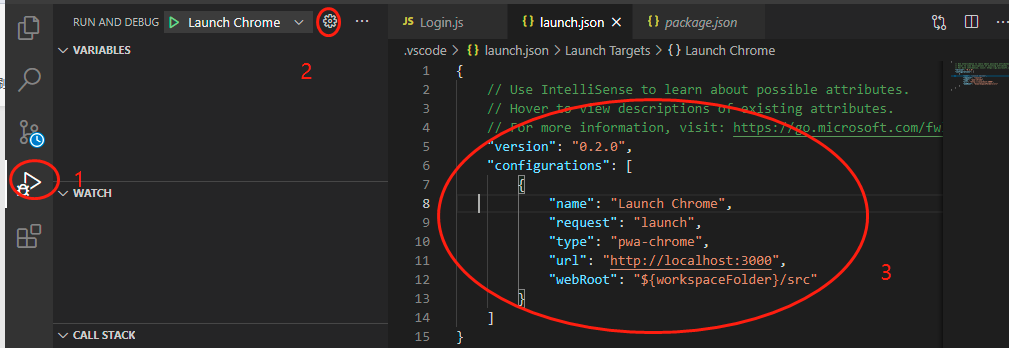
# Debug reactjs component life cycle

Download this repo: <https://github.com/onebit256/react-demo>

## Step 1: add scripts to package.json

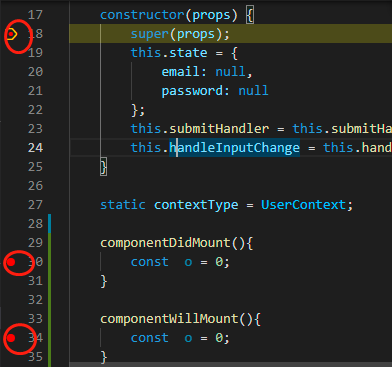


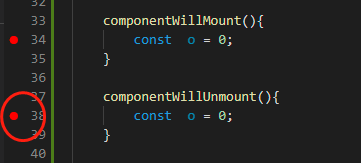
## Step2: config launch.json



## Step3: npm start

## Step 4: put break points on source/component/login







## Step 5：click 2 launch in step 2

## Step 6 start debugger

go to browser and type 127.0.0.1:3000/login to trigger the debugger

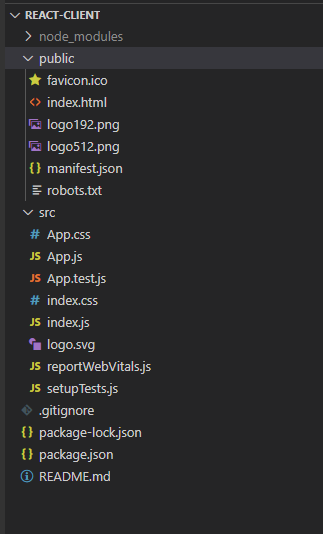
You will be able to see the component life cycle

# Transform html to Reactjs

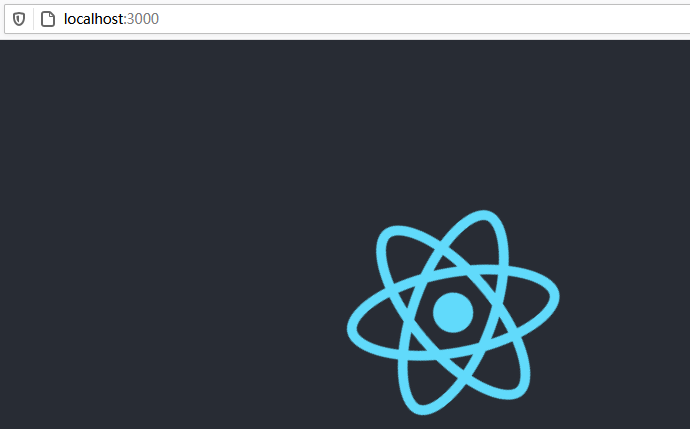
## Step1: init project

npx create-react-app .

This will auto generates the folder structure

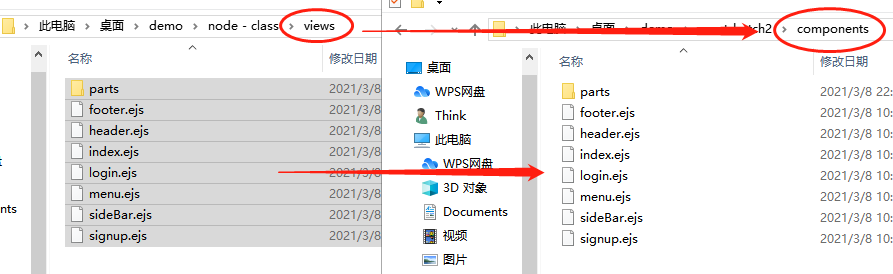


Npm start to test reactjs, you will see

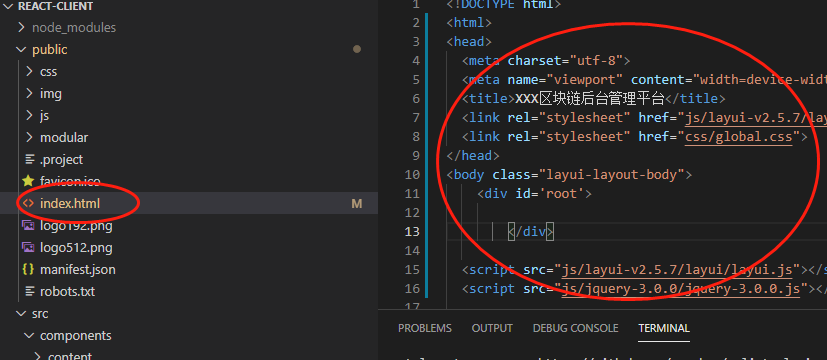


## Step2 convert all ejs template to class components

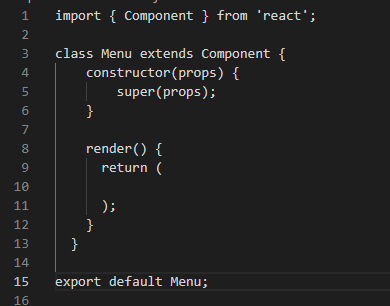
1. Creating empty folder called components under src folder
2. Copy all the ejs from lab5 to components folder



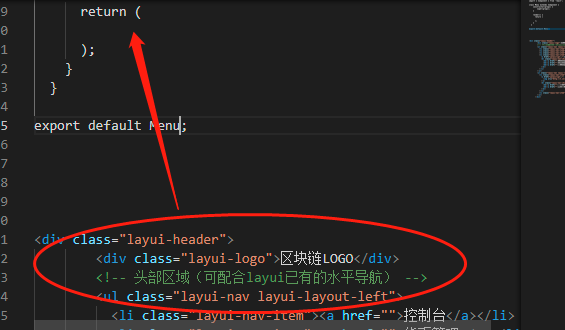
1. copy header.ejs and footer.ejs to index.html, write it as follow



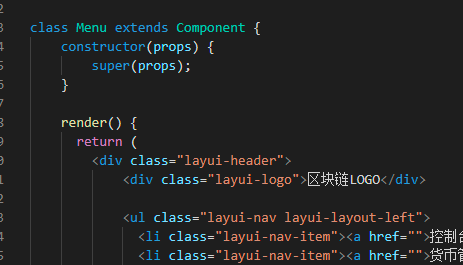
1. Convert ejs to react component: copy and past this template to each file and change the file extension to js



1. Convert ejs to react component: move the original html to return

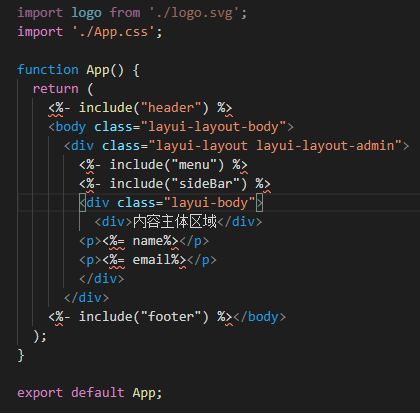


And it becomes

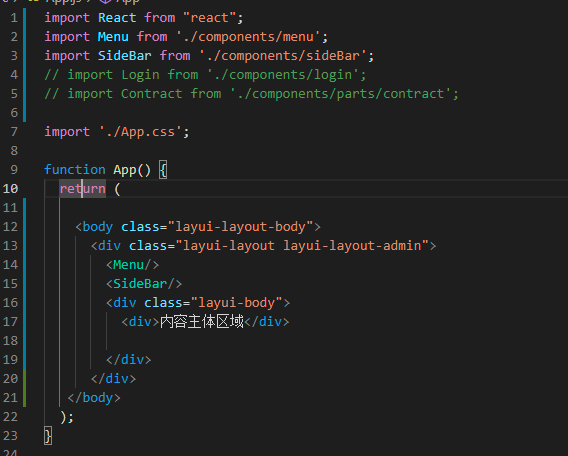


Do the same thing for sideBar Login and signup

For index.ejs, move it to app.js, and change include menu to menu component



To



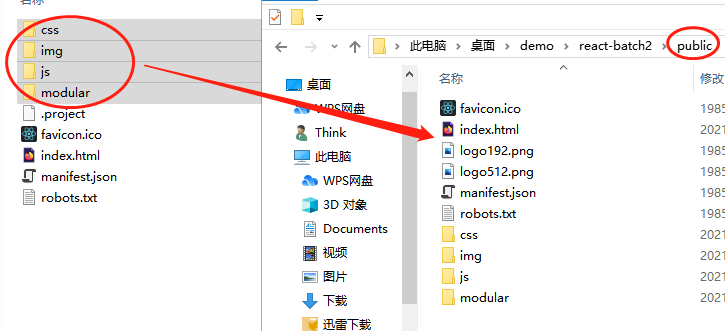
Then npm start

And go to 127.0.0.1:3000, you will see

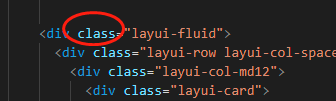


What is wrong with this styling?

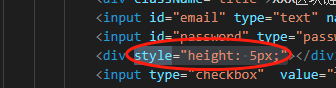
## Step 3: add css and js to public folder



## Step 4: replace all class= to className=

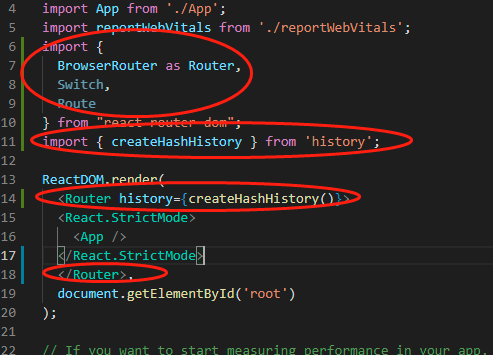


## Step 5 remove all style



# 3. Add router

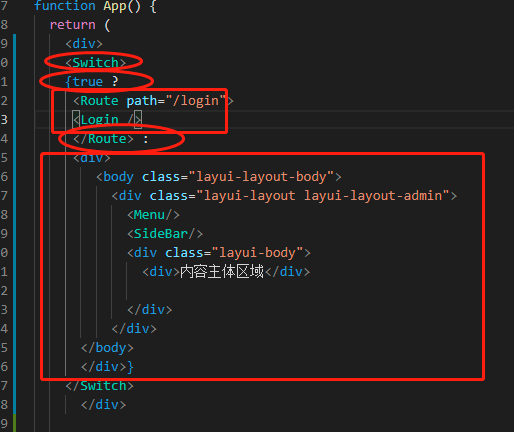
## Step1: add route wrapper to index.js



- history will keep tract of the path.

- Router is a third party context

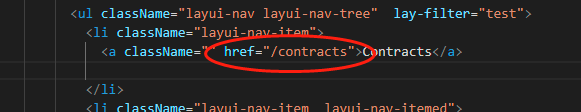
## Step 2: add route switch to app.js



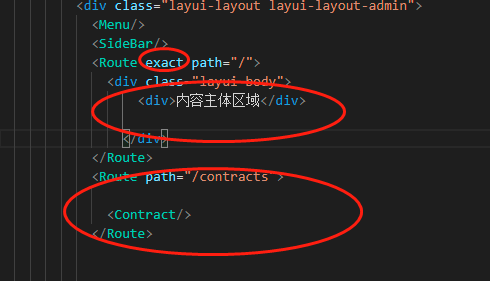
- Switch is analogous to if statement to determine the routes

- if statement in return { condition? Statement 1: statement2}

## Step 3 implement hyperlink in react route

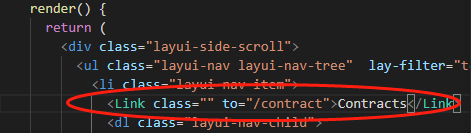


Have to create a corresponding path in the route



- Exact means accurate matching, what would be the problem without exact.

Or we could use Link



# 4. Add Nodejs login to React

## Step 1: how to implement the form

1 form html



- react use events listener to submit the form data

- Define a onSubmit event listener

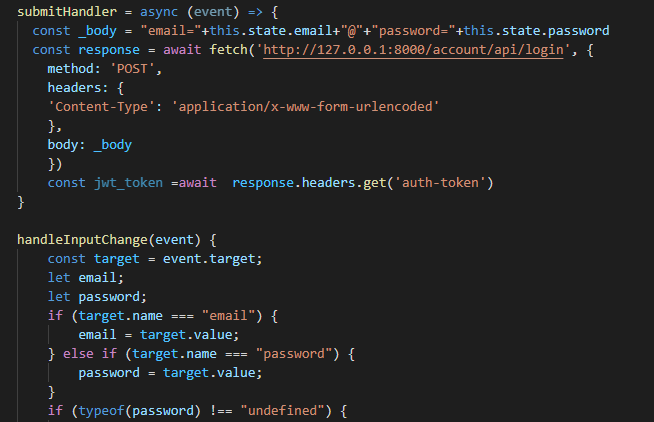
- add event handler function

- react use events listener to store the form data the form

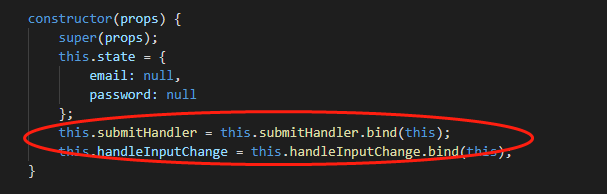
- Define a onChange event listener to handle the input

- add event handler function

2 Add events Handler

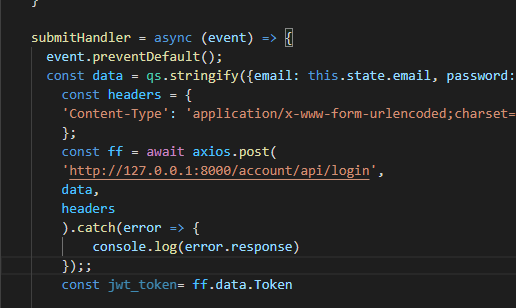


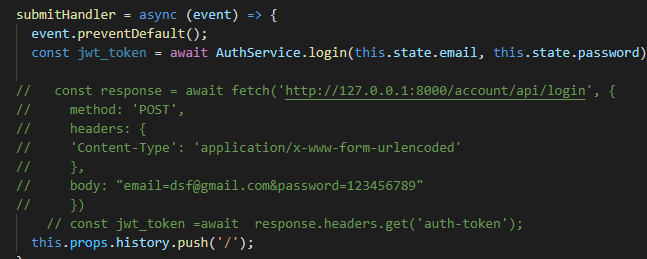
3 Add bind



Why add bind?

## Step2 form handler login - post request



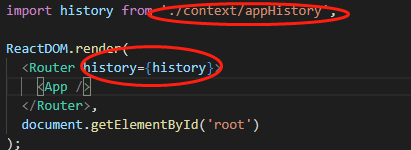


## Step 3 add route context

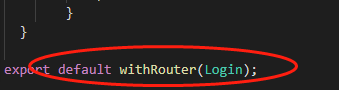
Create



Add



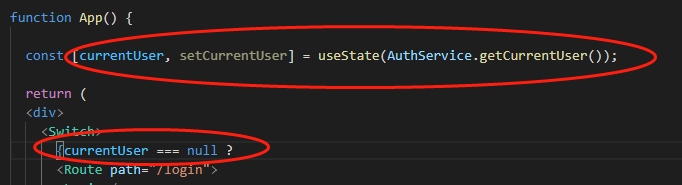
Use context



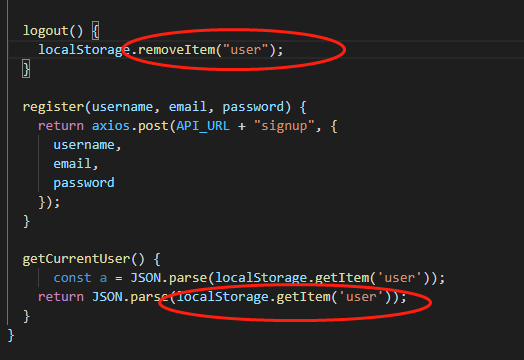
## Step 4 add auth

<https://bezkoder.com/react-jwt-auth/>

Add state to keep track of user login



Add localStorage in auth service



Add logout event handler