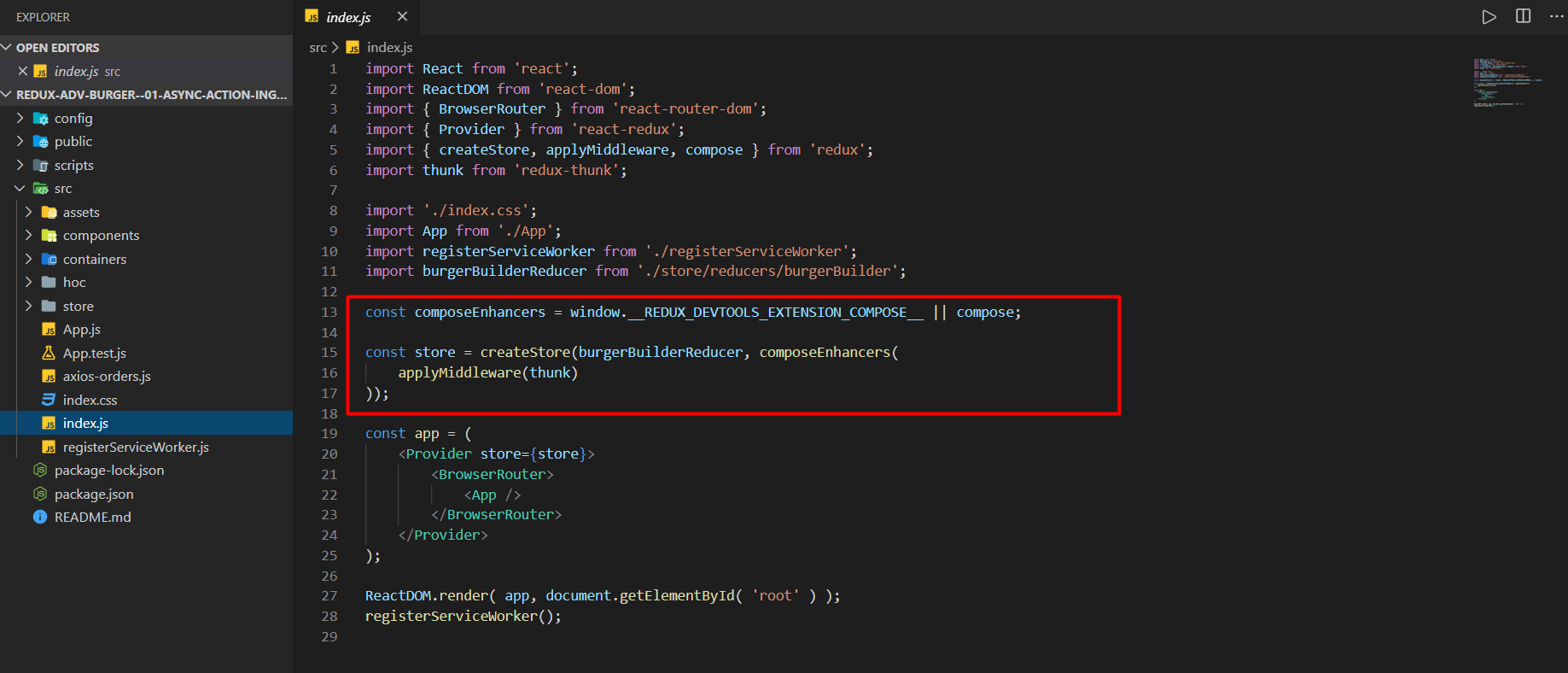
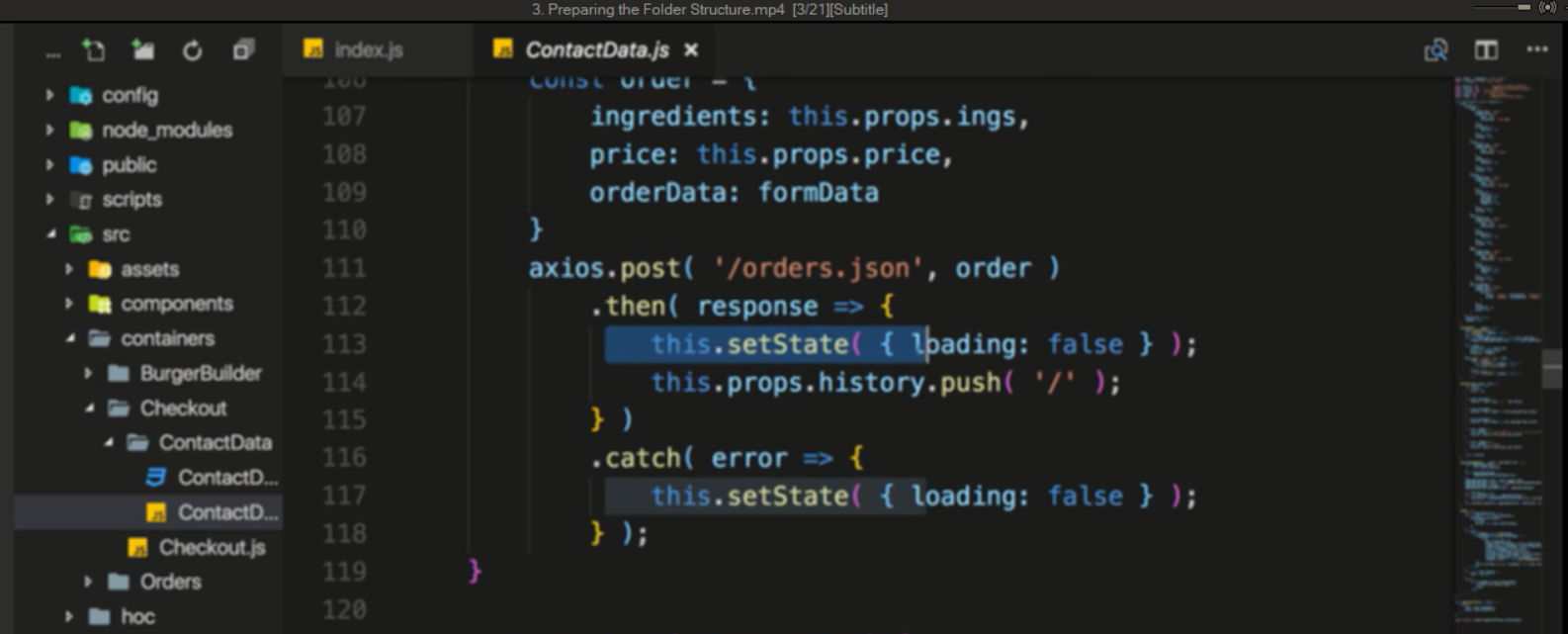
2. Installing the Redux Devtools

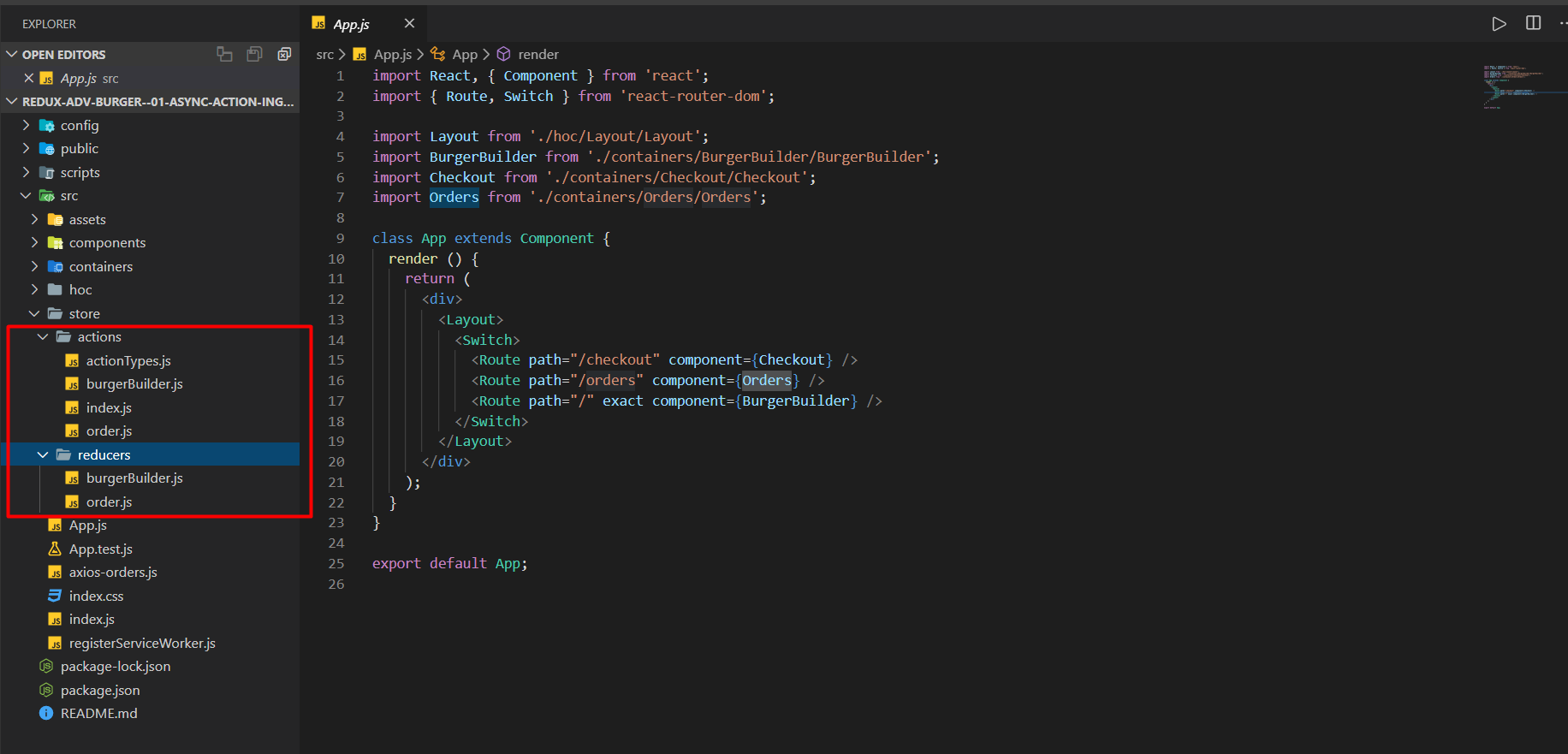


3. Preparing the Folder Structure

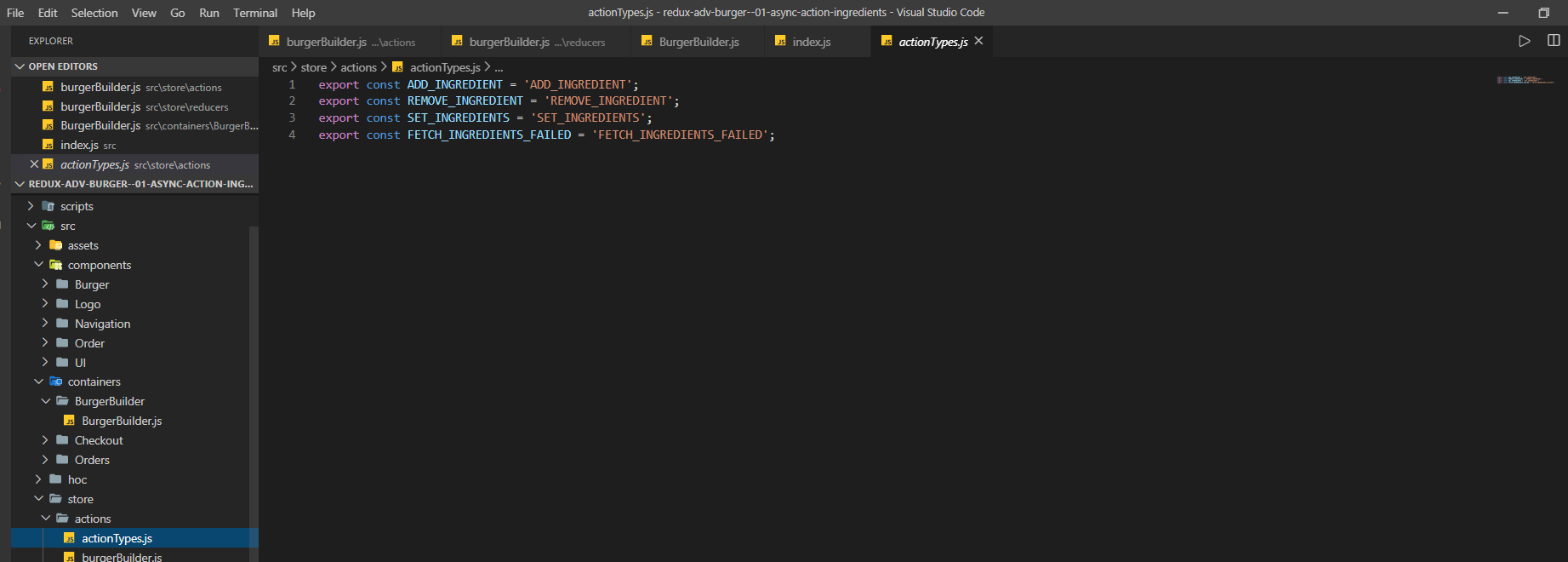


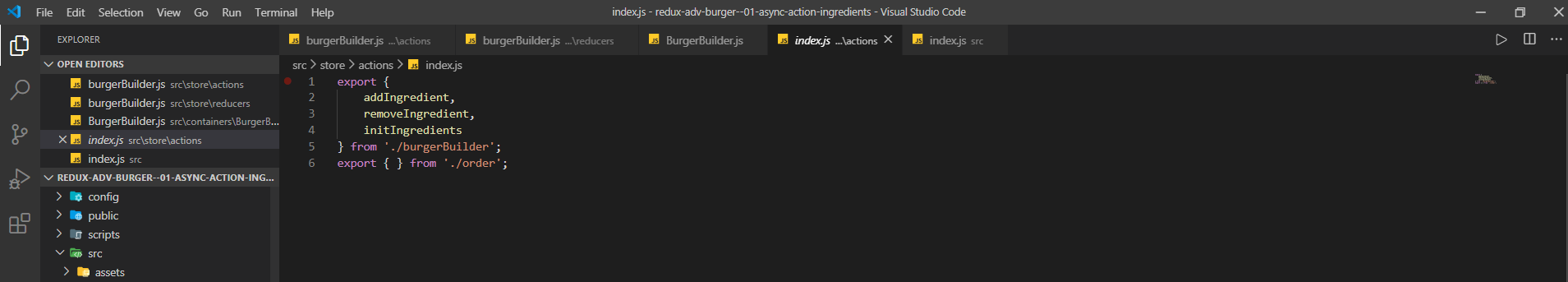
Send order lưu vào db

* Chuyển cái này sang dùng redux ( redux with async)

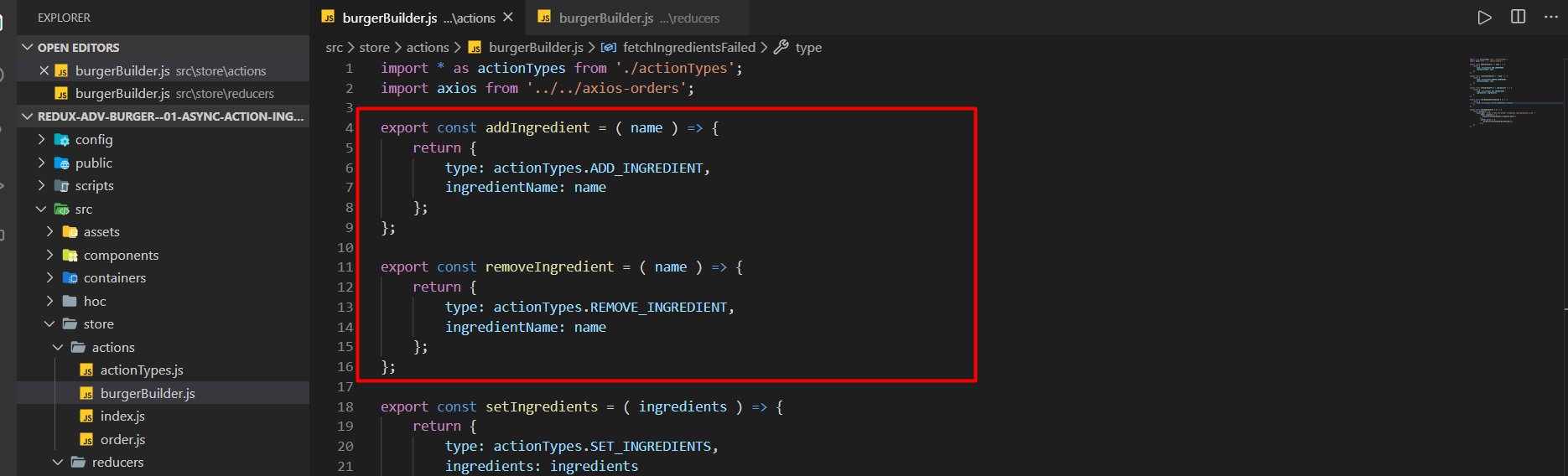


Tạo các thư mục cho redux

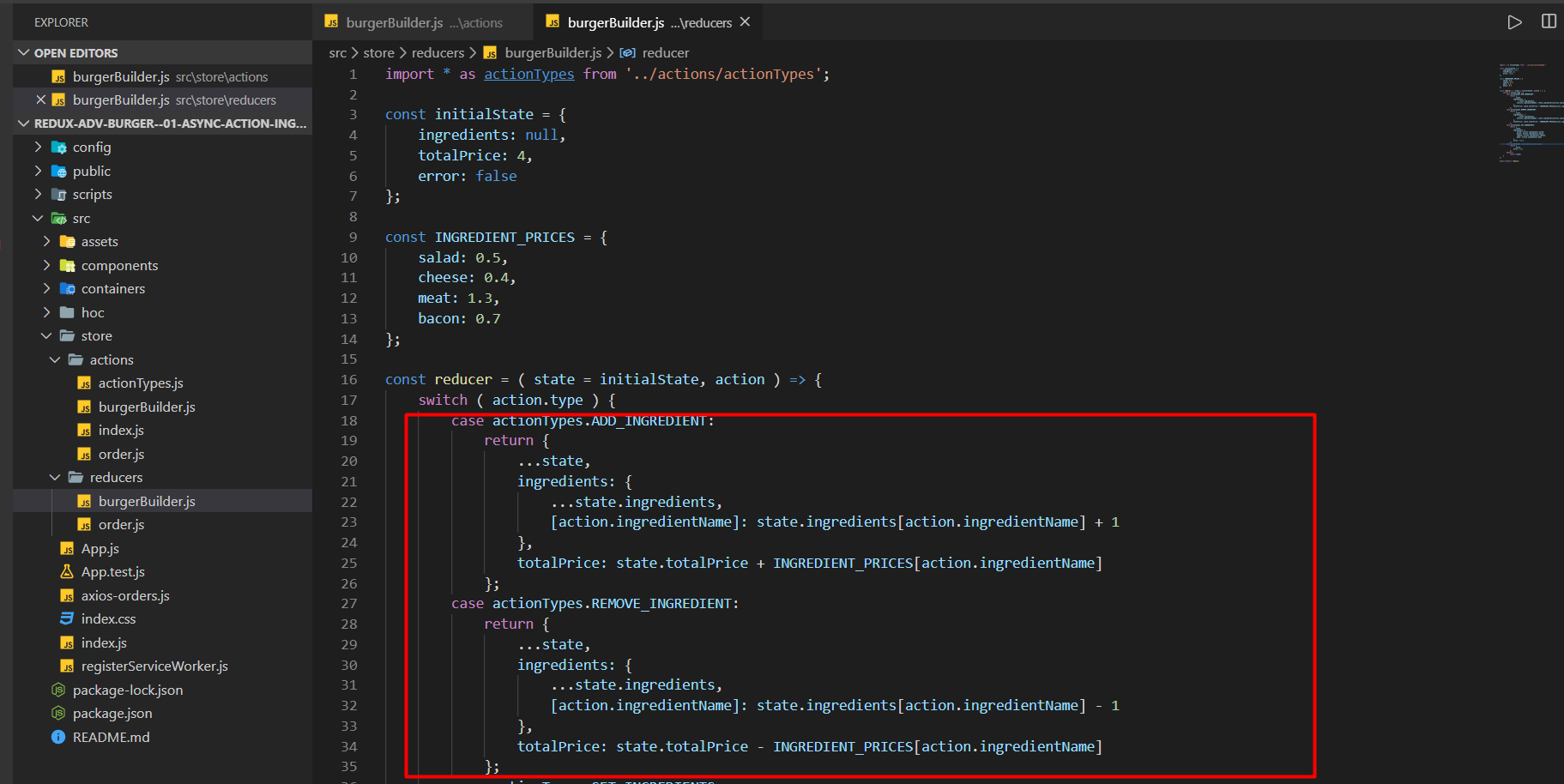




4. Creating Action Creators

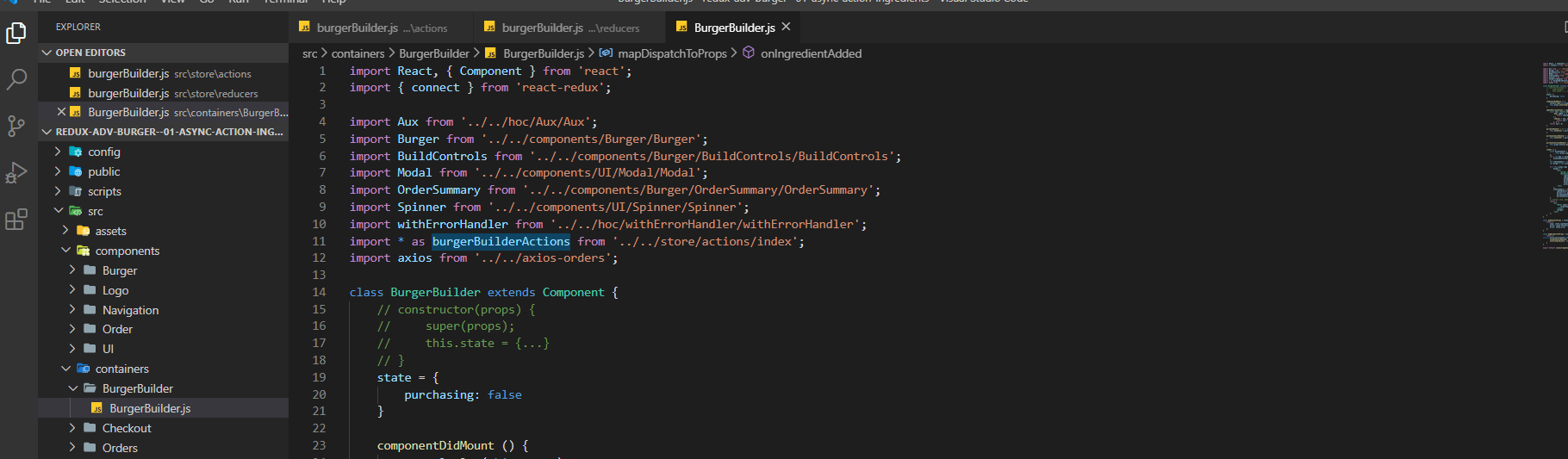


Action

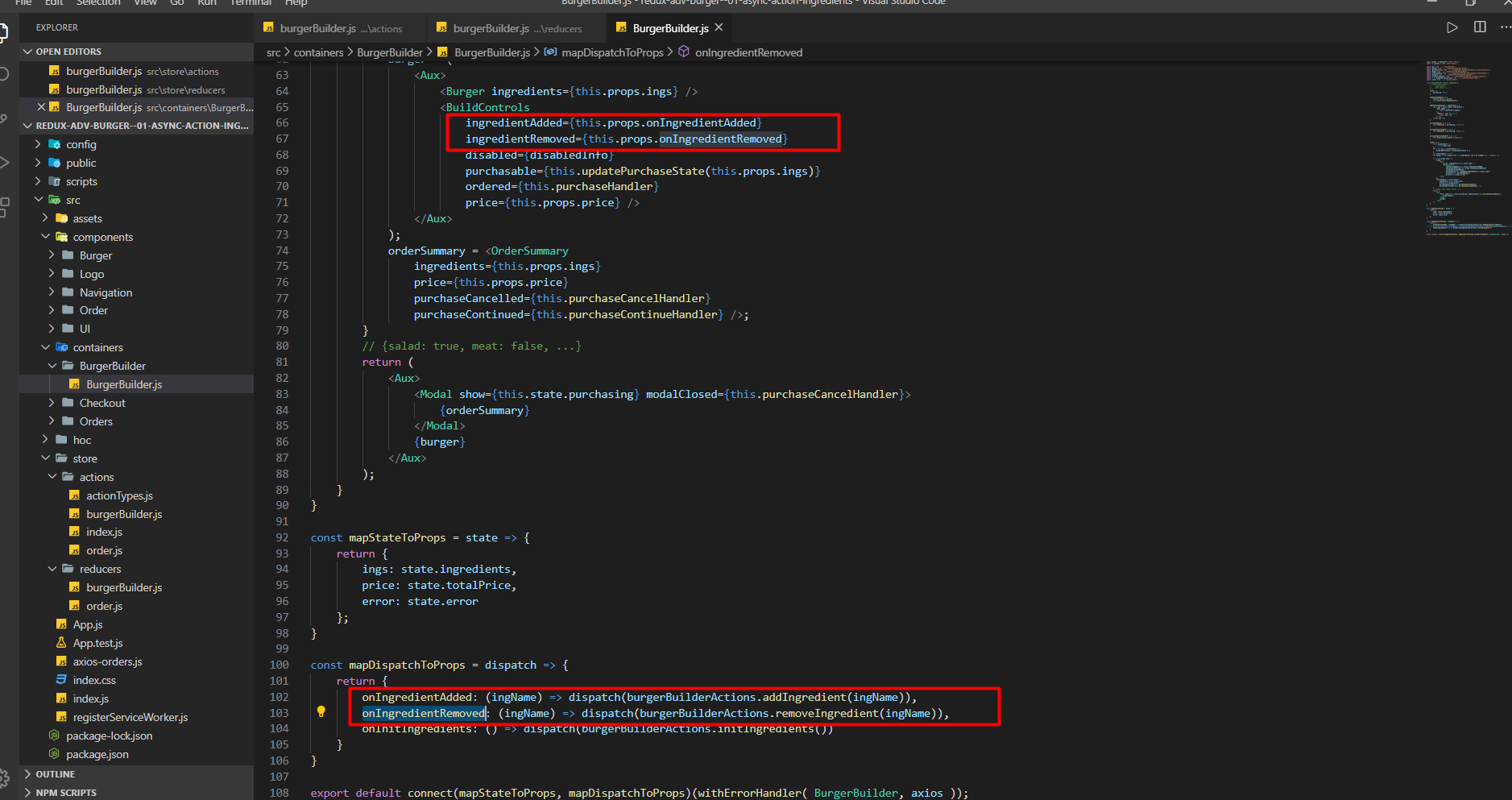


Reducer

Việc tính toán lại price được thực hiện tại action add và remove ingredient luôn, nên đứa ingredient price vào đây luôn



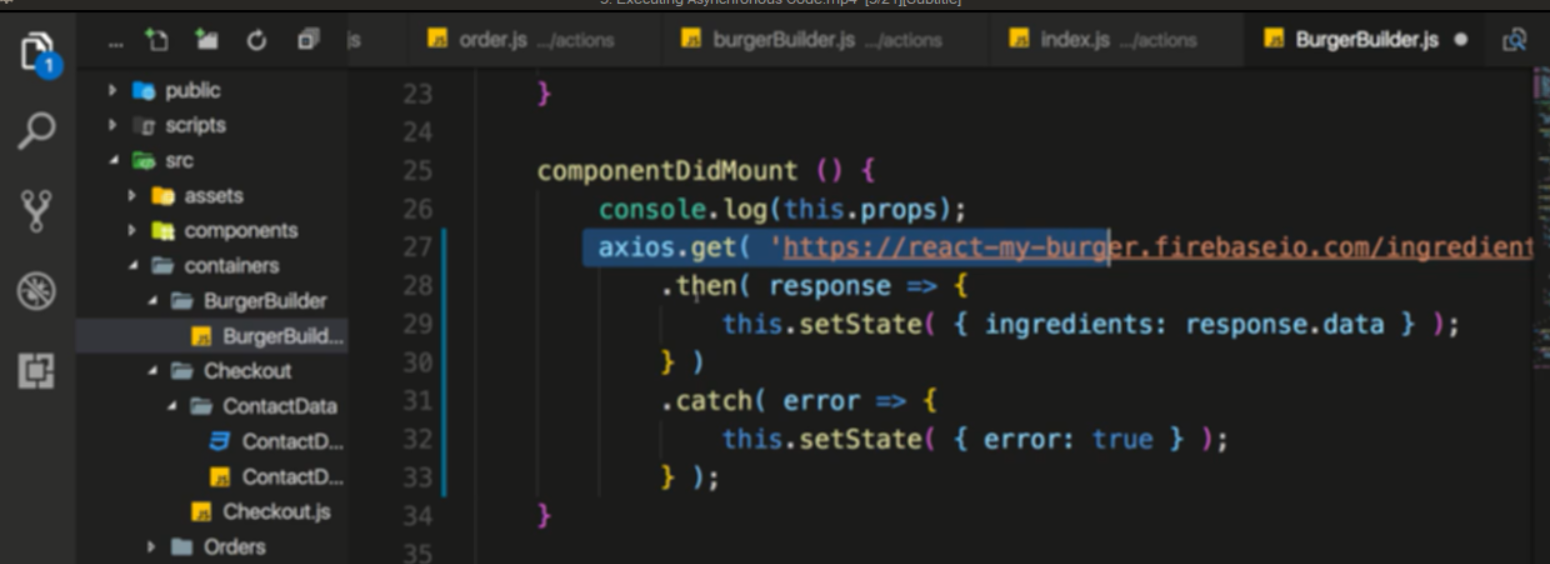
Import các action vào burger builder



Map dispatch và sử dụng trên burger builder

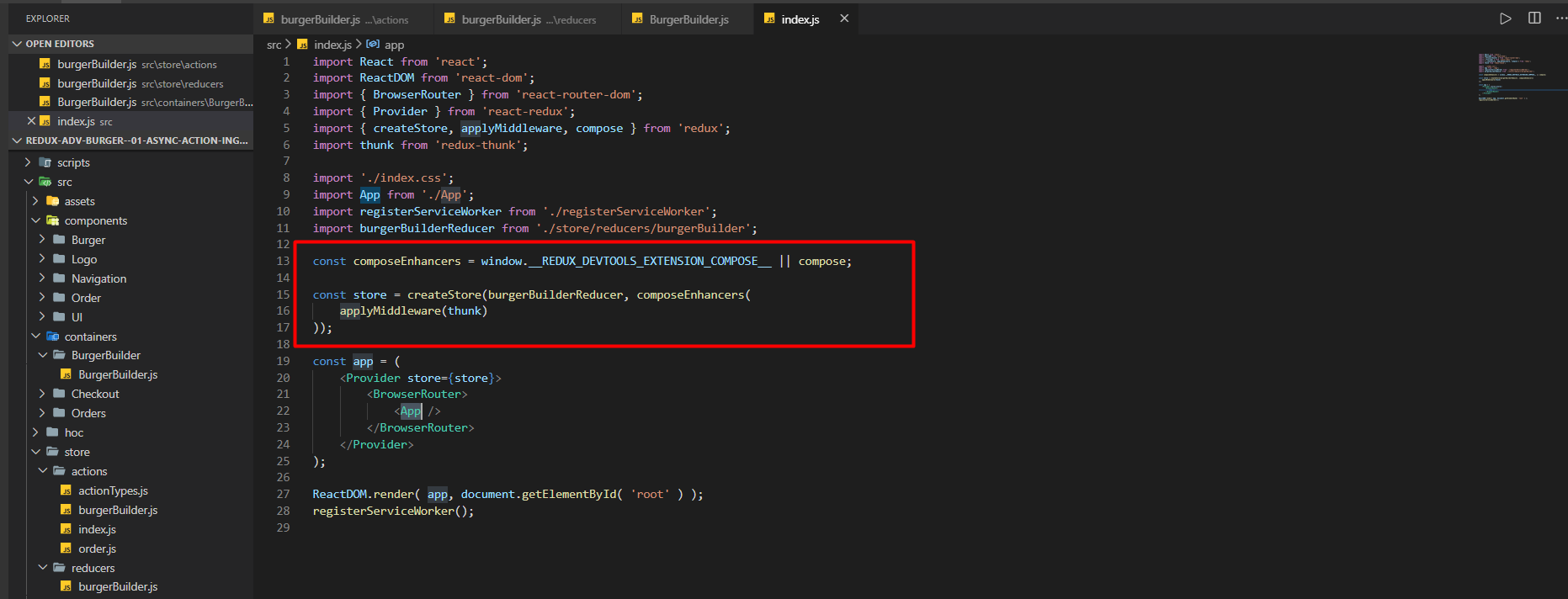
Trong hàm map dispatch ở đây là định nghĩa hàm nên có biến định nghĩa trong đó, trường hợp refer hàm thì không cần đưa biến vào ()

5. Executing Asynchronous Code



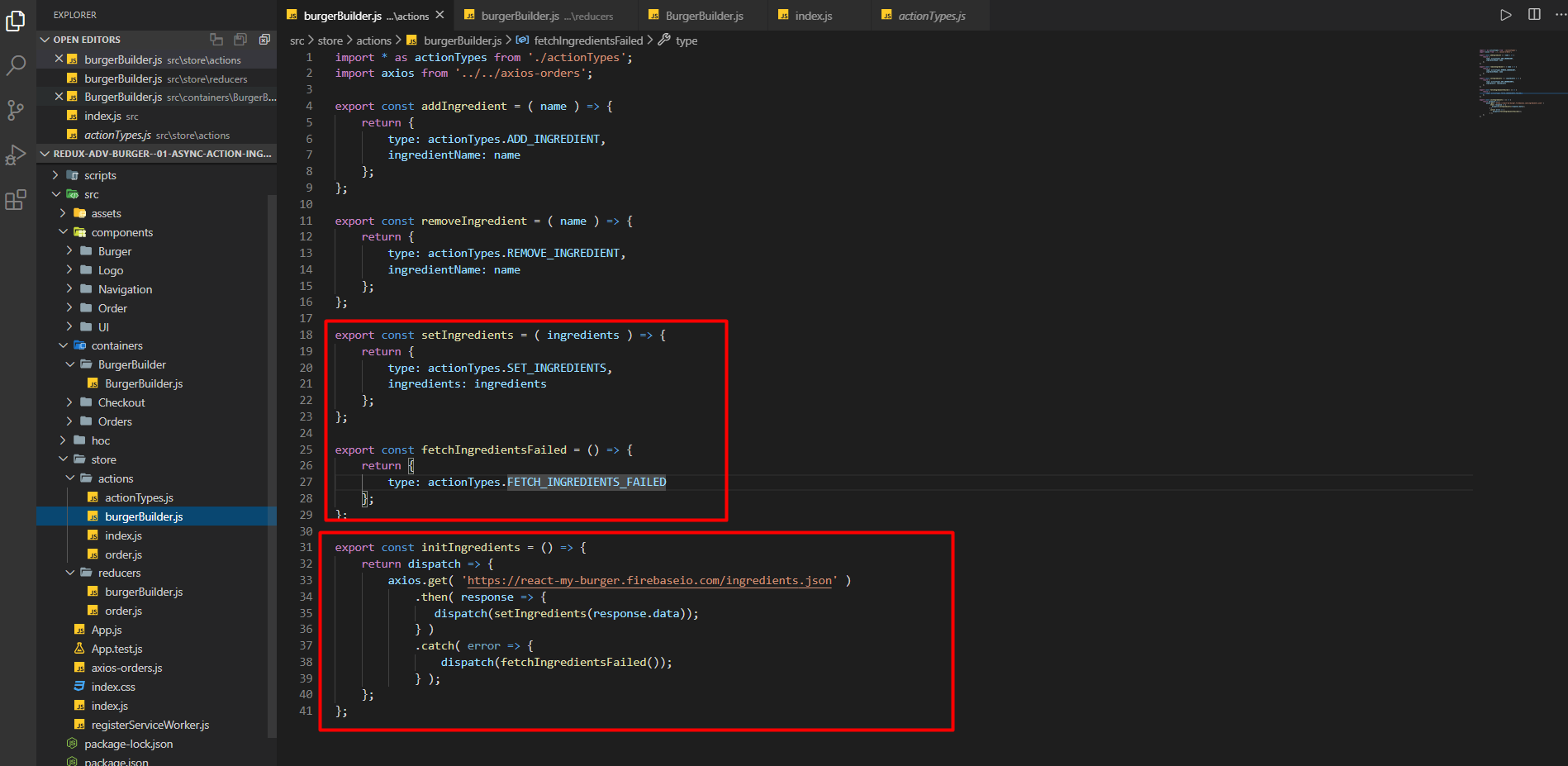
Đưa phần lấy ingredient đưa vào state khi didmout vào sử dụng trong redux

Dùng async thì phải install redux thunk



Đưa redux thunk vào

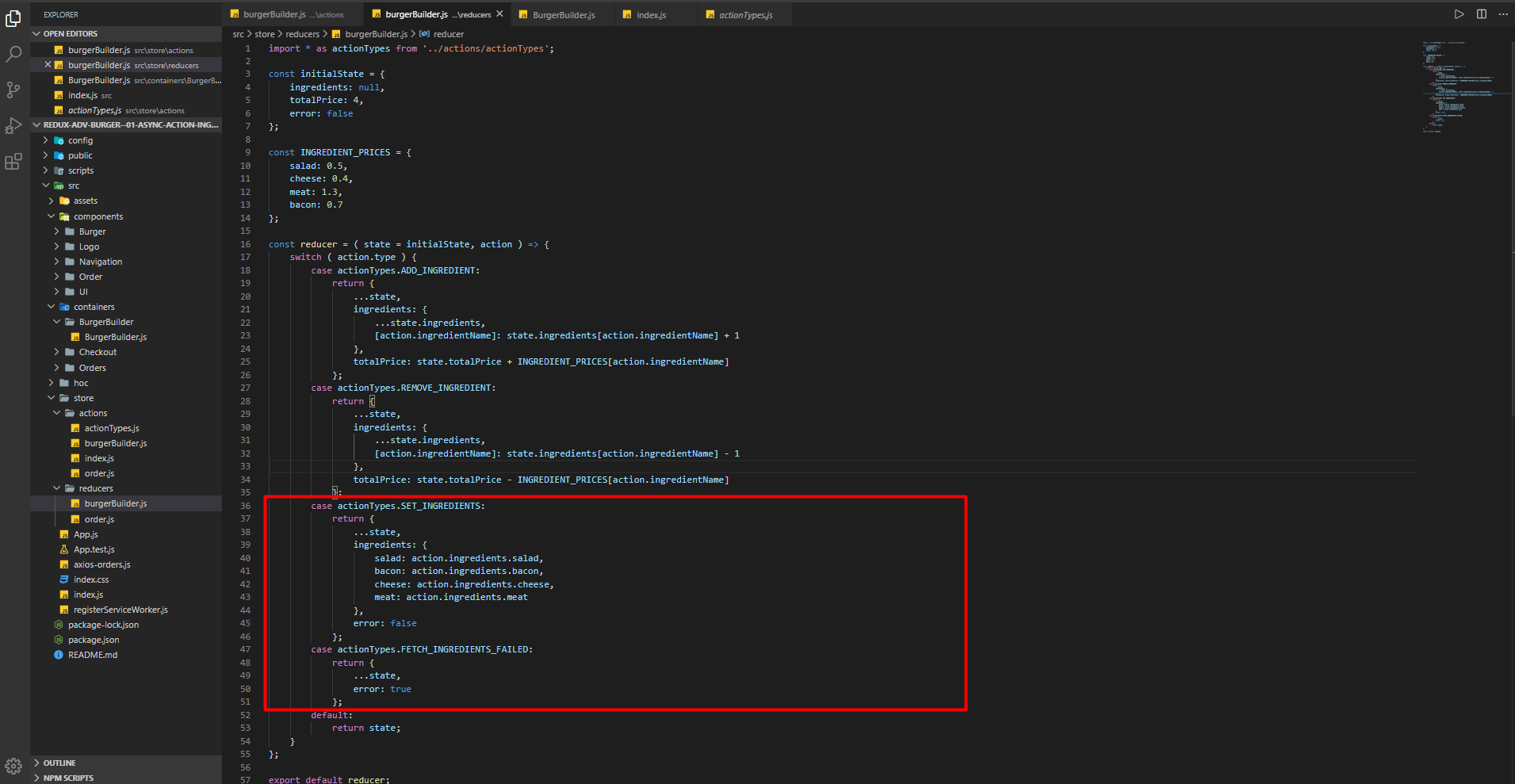
5. Executing Asynchronous Code



Action

Khi axios thành công thì gọi action setIgredient

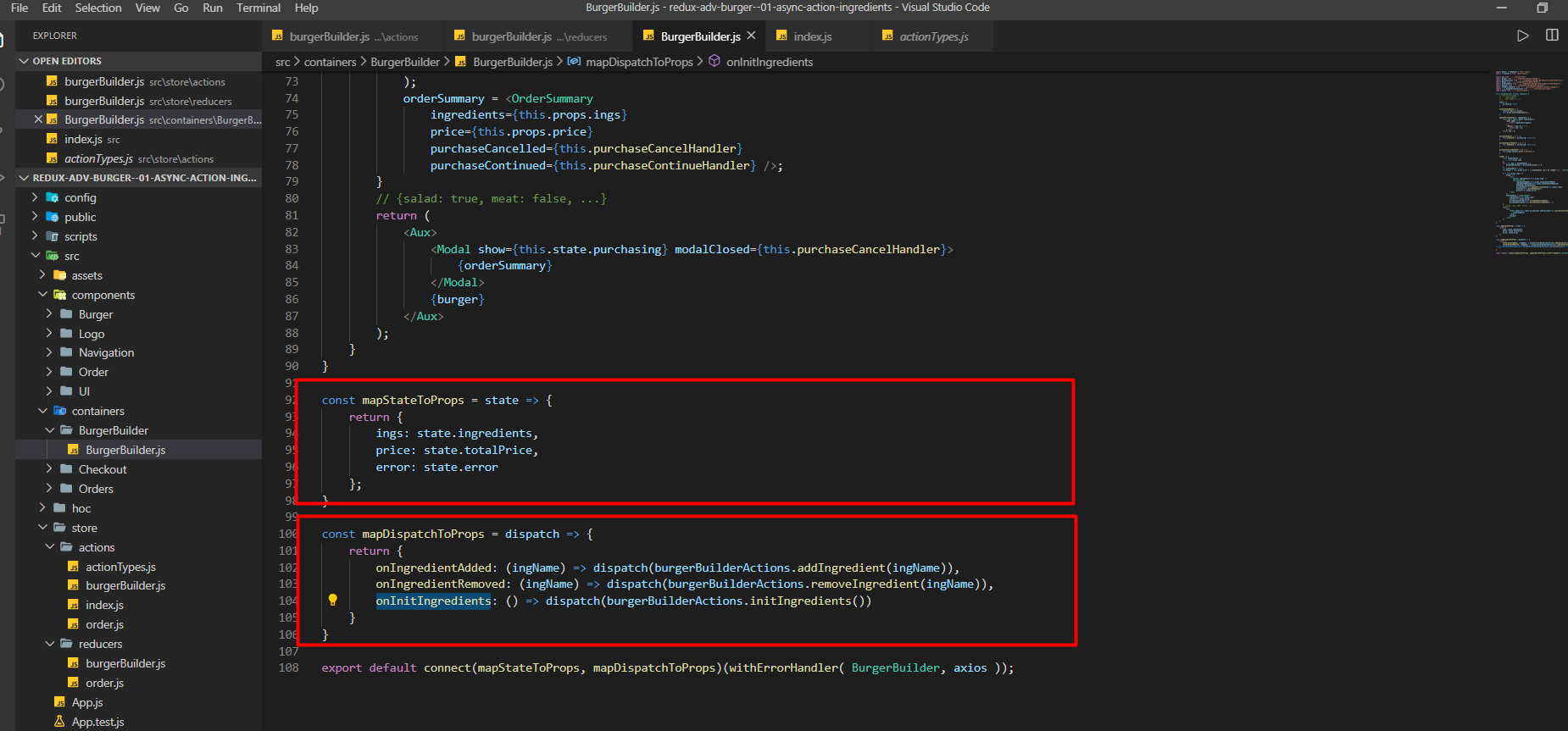
Khi axios nhảy vô catch thì gọi action fetchIgredientfailed



Trên reducer, set state ở trường hợp axios thành công và trường hợp false

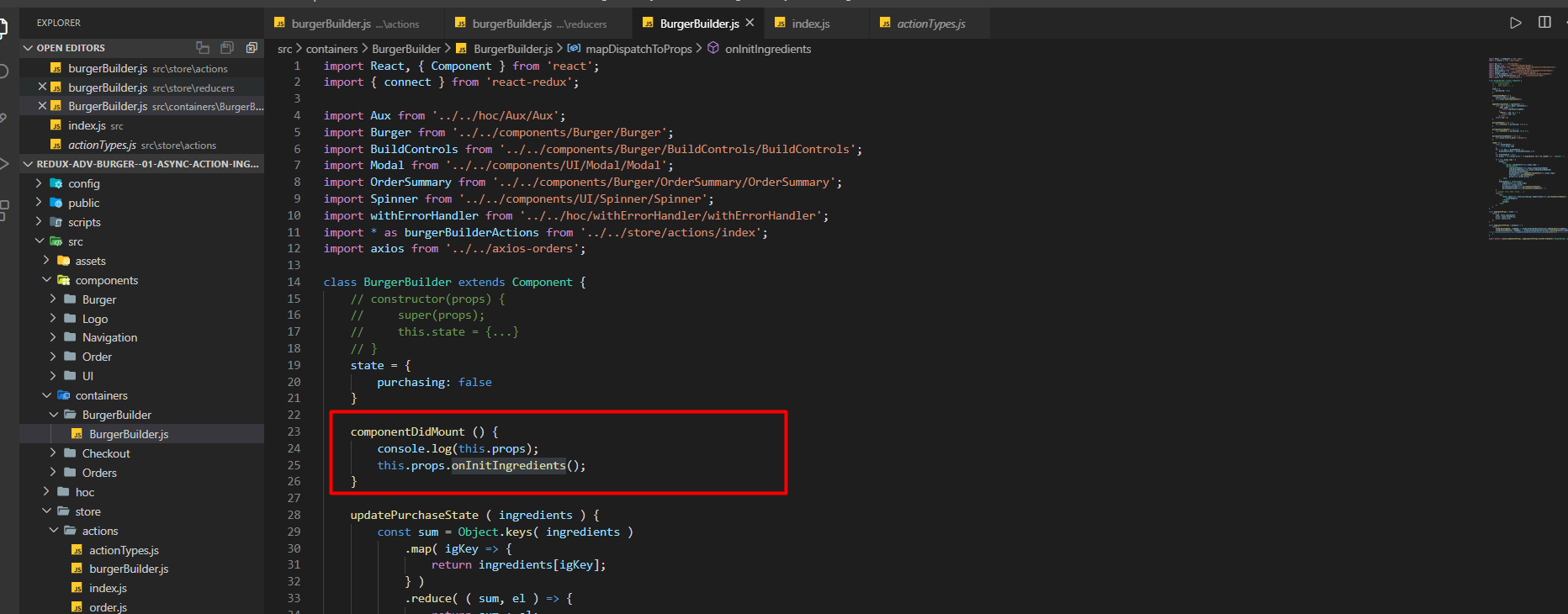
Nhớ thêm các state mới vào initialState như error

Ingredients chỉ cần action.ingredients là lấy được, như đi vào trong sald,bacon,cheese, meat thì có thể thay đổi thự tự của các ingredient trong object => các lớp của cái burger được thay đổi



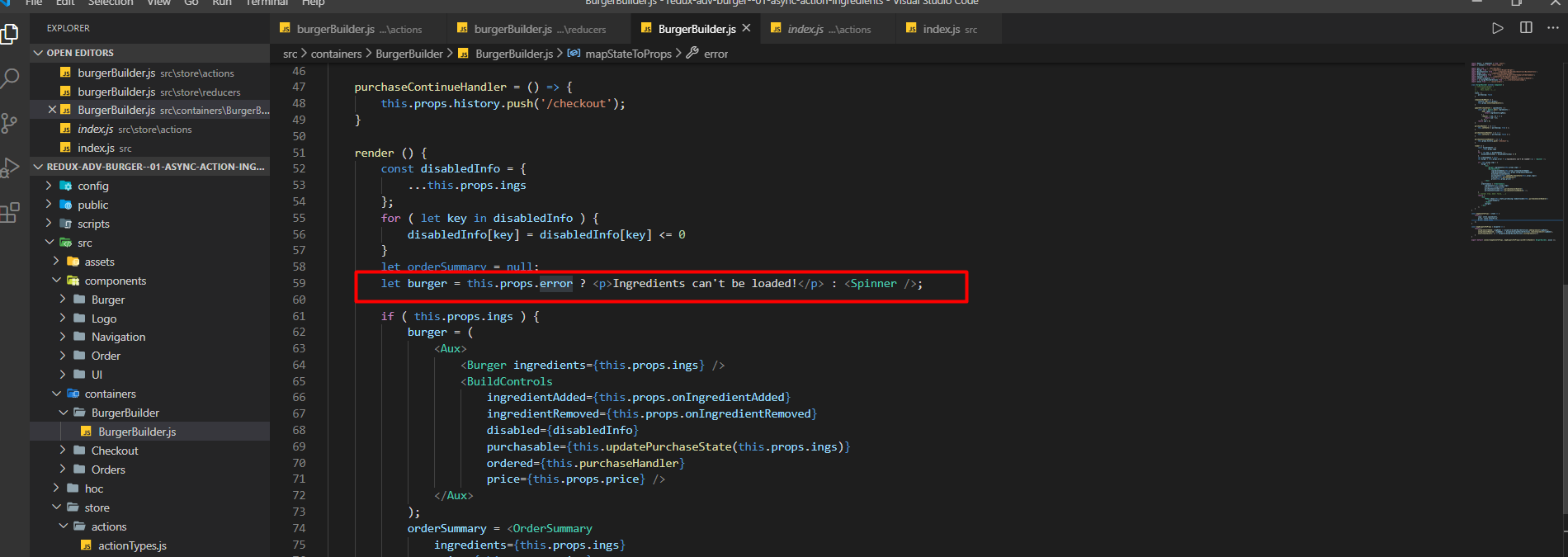
Mapto props

Và state có thêm error ( khi gọi axios false thì error sẽ được set lên truy)



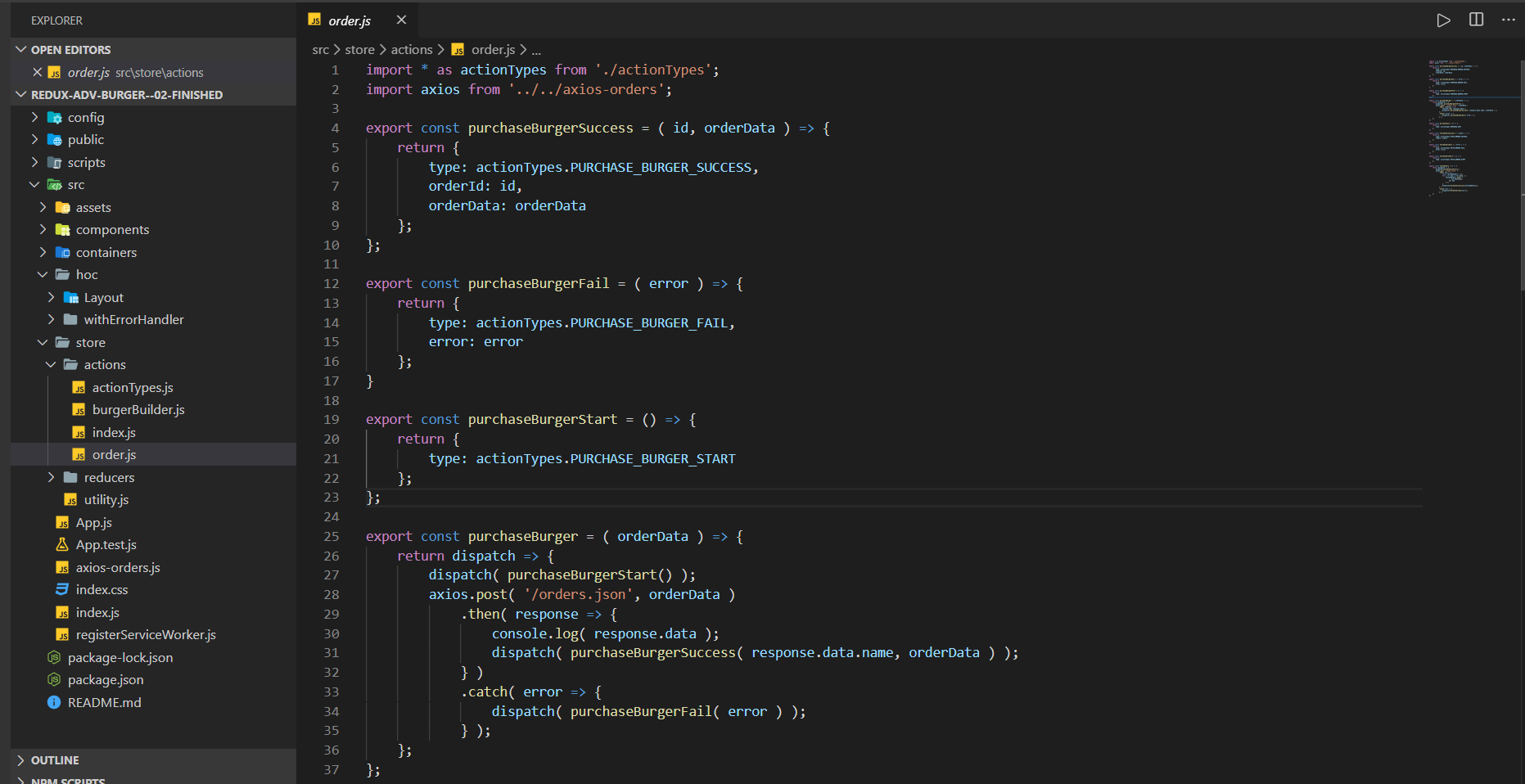
Xài trên did mount để lấy ingredient từ server về và set vào state

Nên xài redux khi hàm lấy ingredient cần xài trên nhiều component, còn không thì xài local thôi cũng đk



Dùng biến error

9. Adding Order Actions



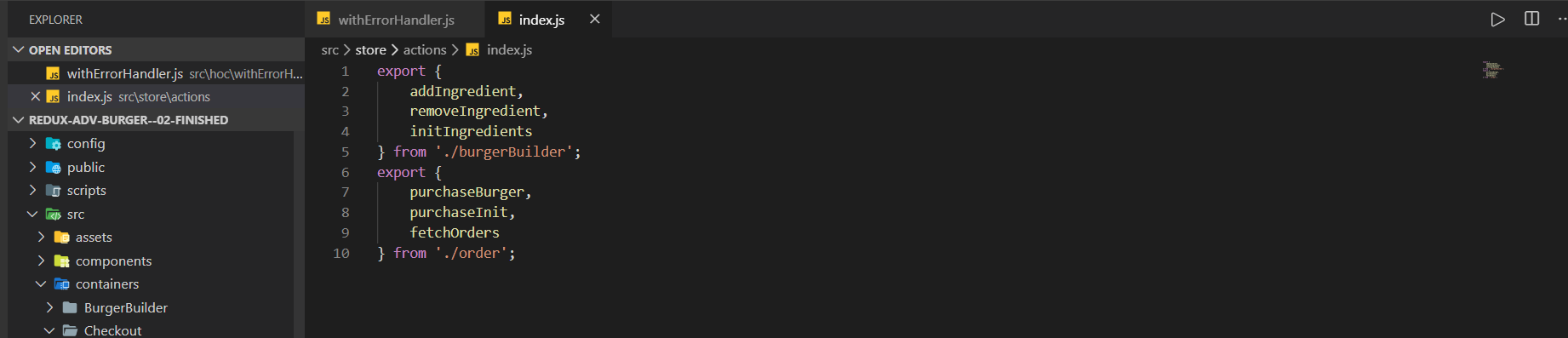
Handler order trong redux

Tạo các action cho việc post order lên server

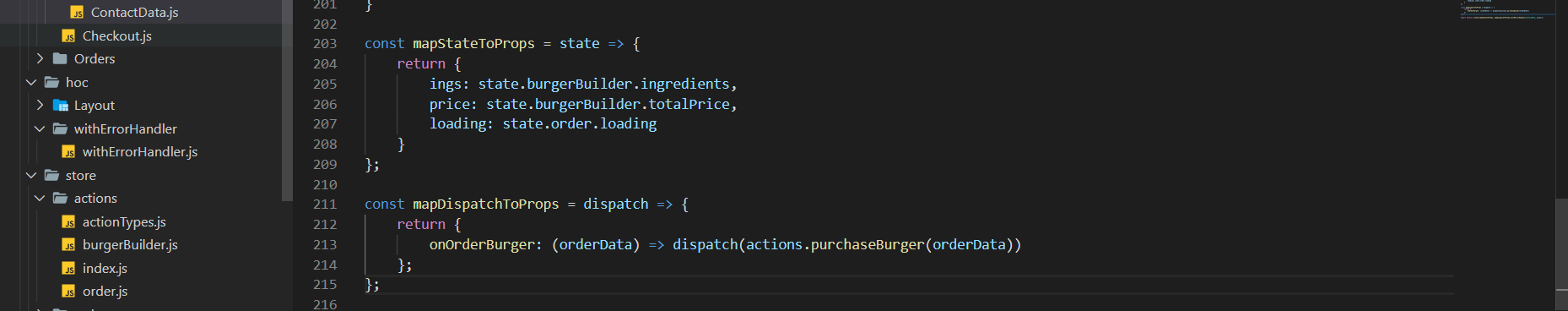


Khi có thêm redux thì mọi thứ được wrap trong redux

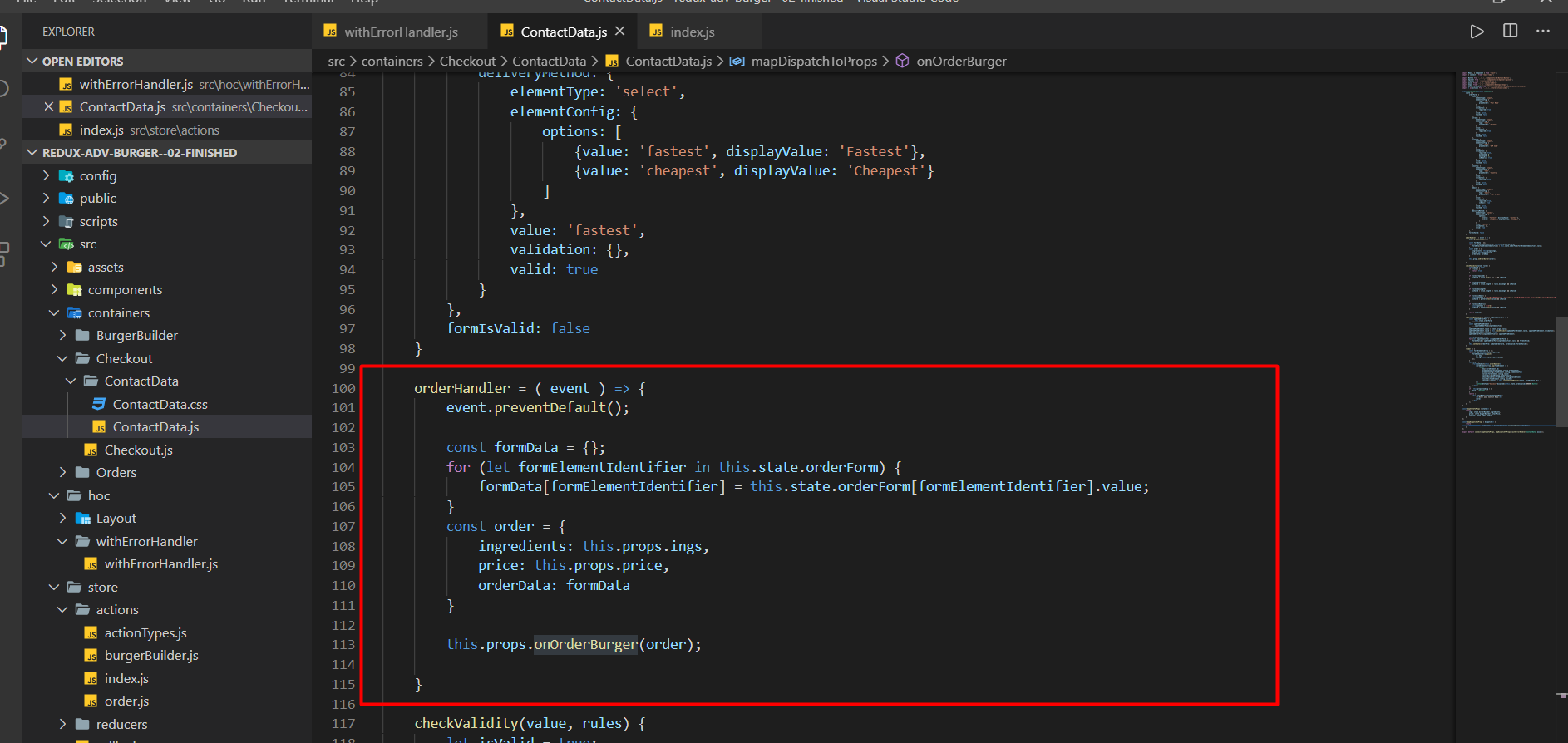
Xem lại cái withErrorHandler trong này



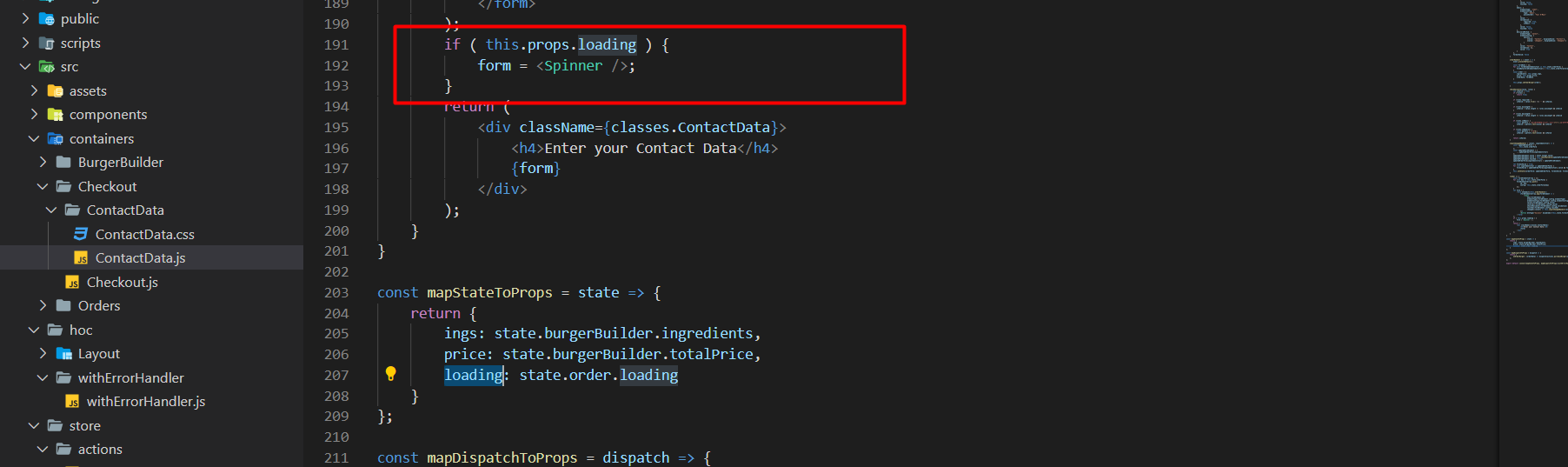
Action/index.js



Map to props

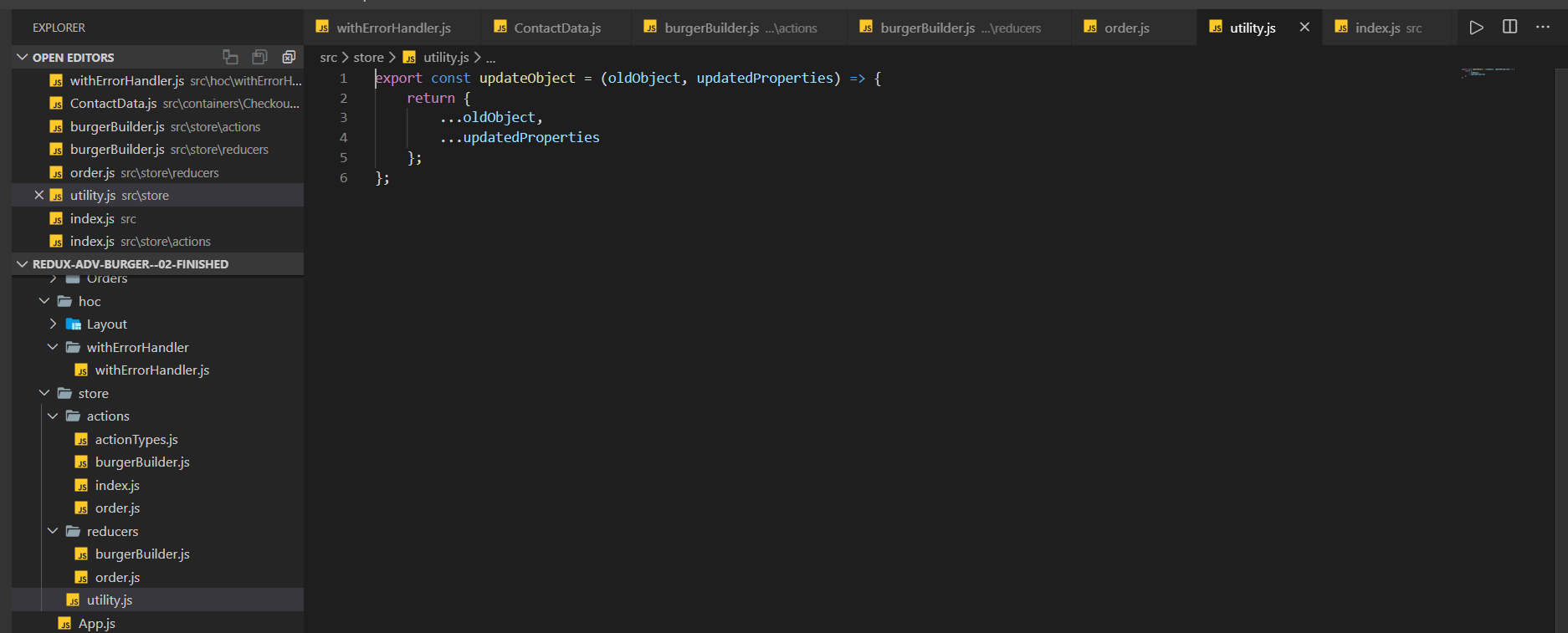


Sử dụng trong hàm orderHandler

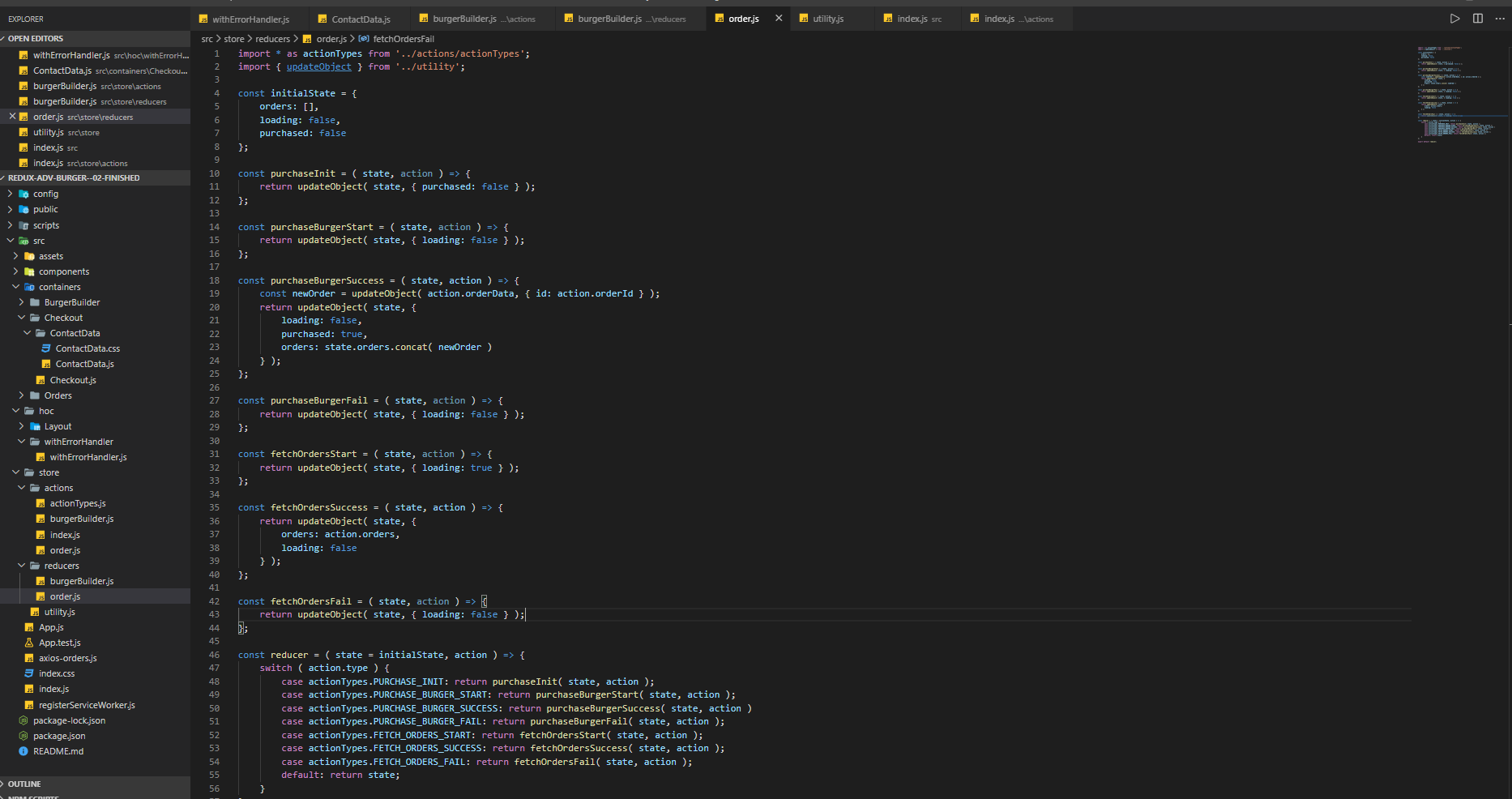


Sử dụng biến loading trong redux state

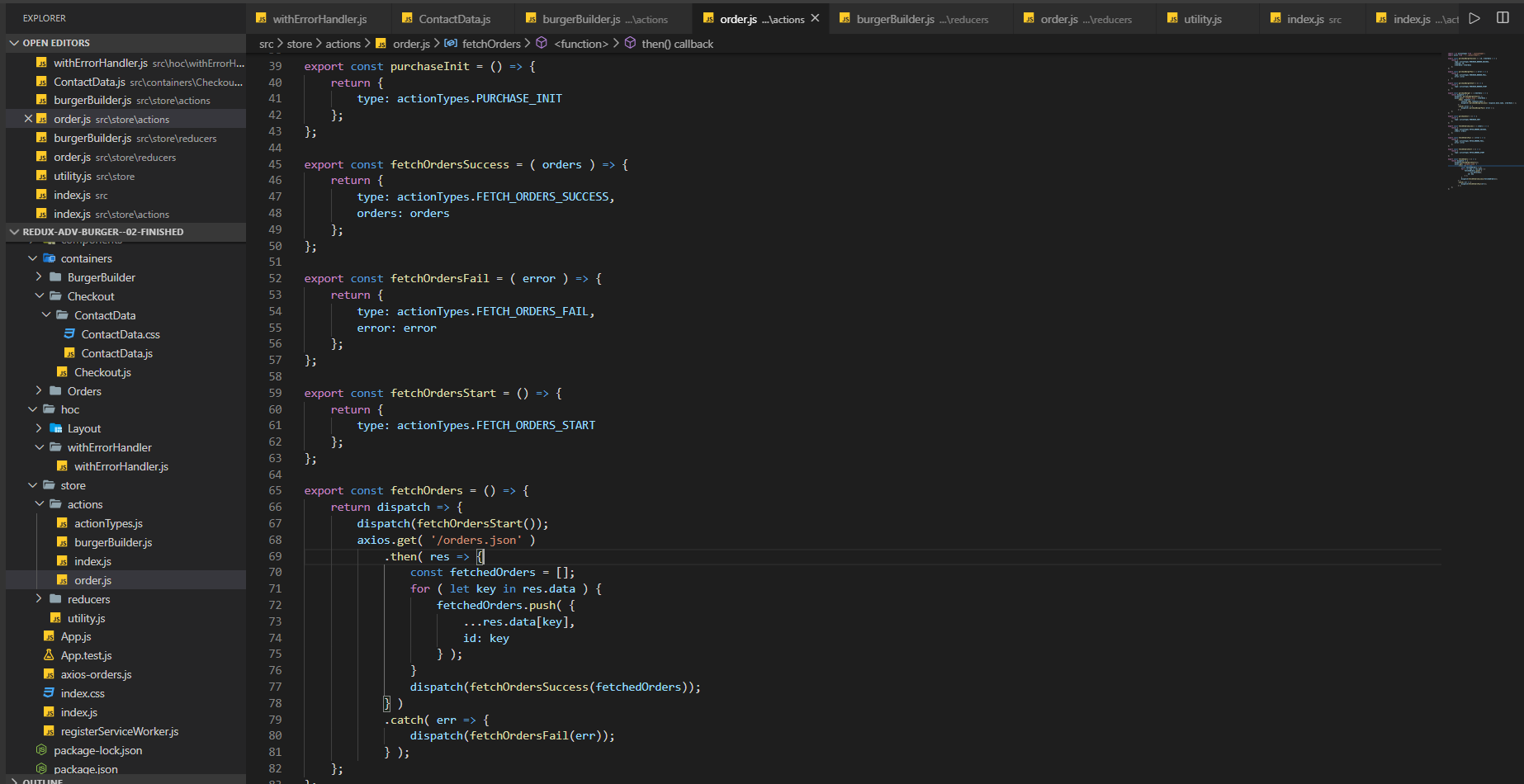
11. The Order Reducer



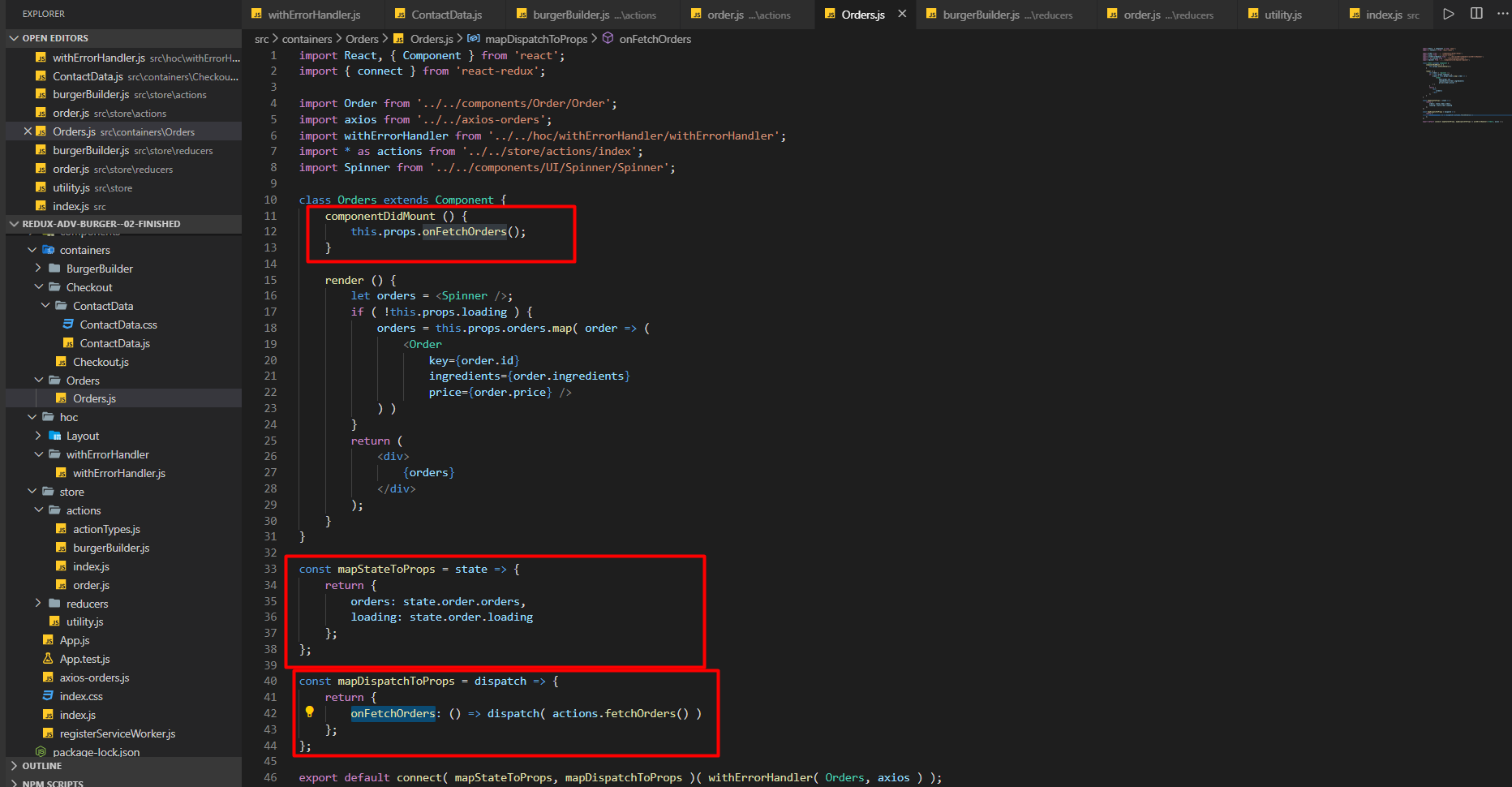
File ultilities



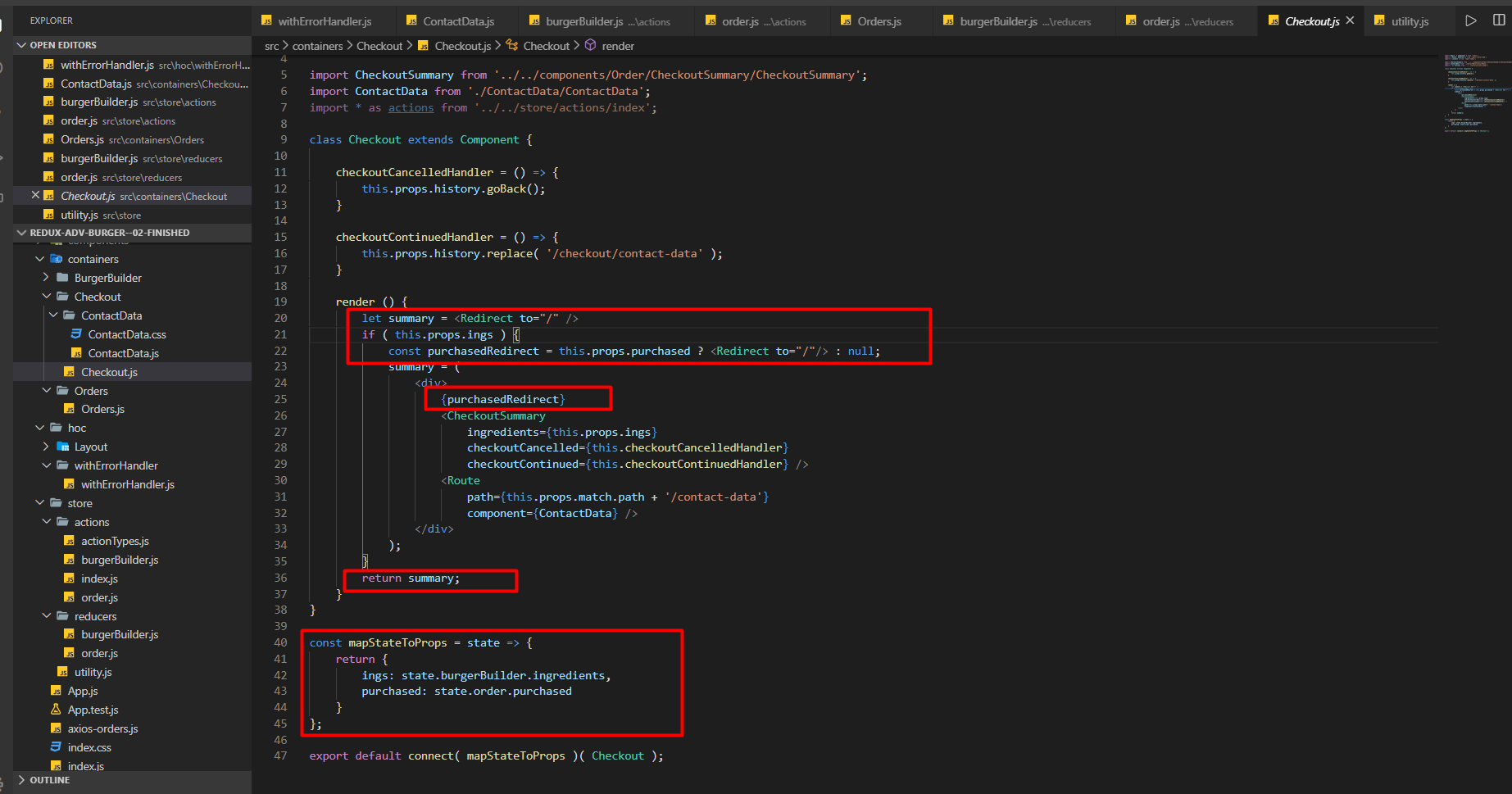
Order reducer



File action của order để thực hiện get các order



Get các order trên component

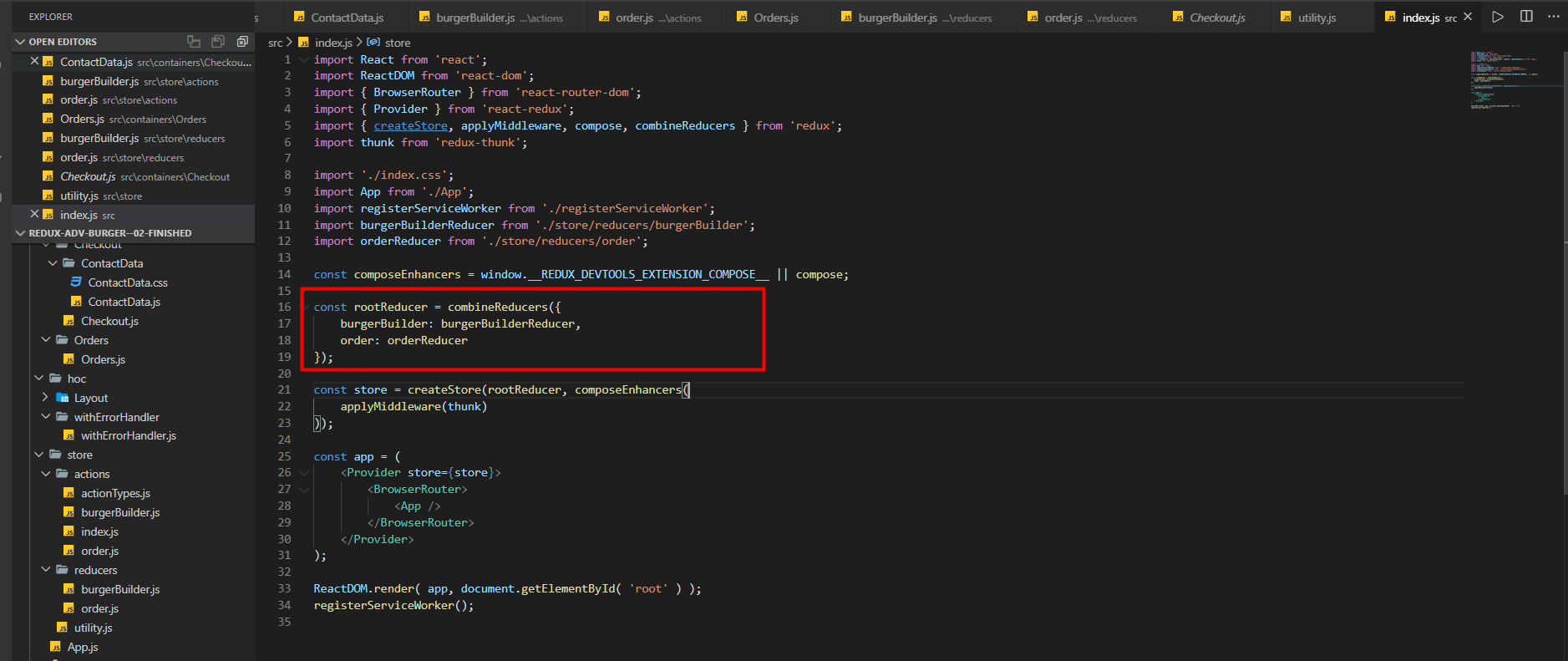


Treen component check out

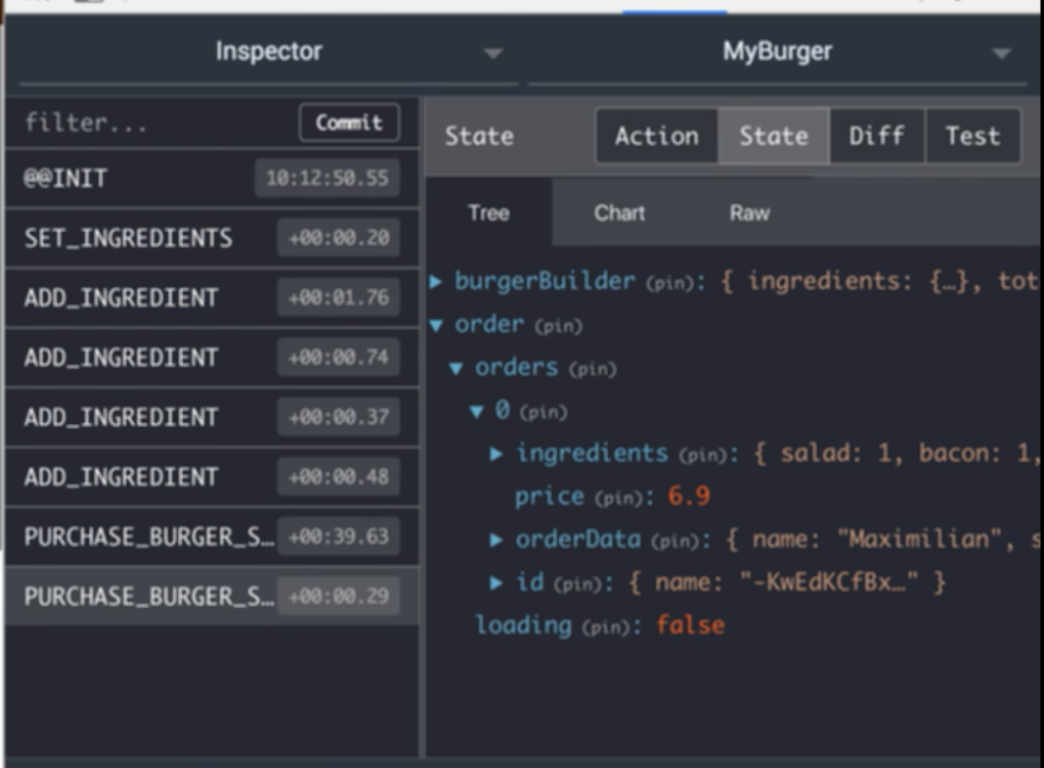
Check xem nếu không có ingredients thì redirect về root, nghĩa là đi từ page add ingredient qua thì lúc đó có ingredient truyền từ redux qua thì sẽ hiển thị trang này, còn dùng link bấm cái bụp thì redirect về root

Khi mua xong thì biến purchased được set true-> redirect về root

Còn lúc ấn hiện lên sẽ hiện componen checkoutSumary và component trong route



Phân ra các reducer rồi dùng combineReducers để đưa nhiều reducer vào

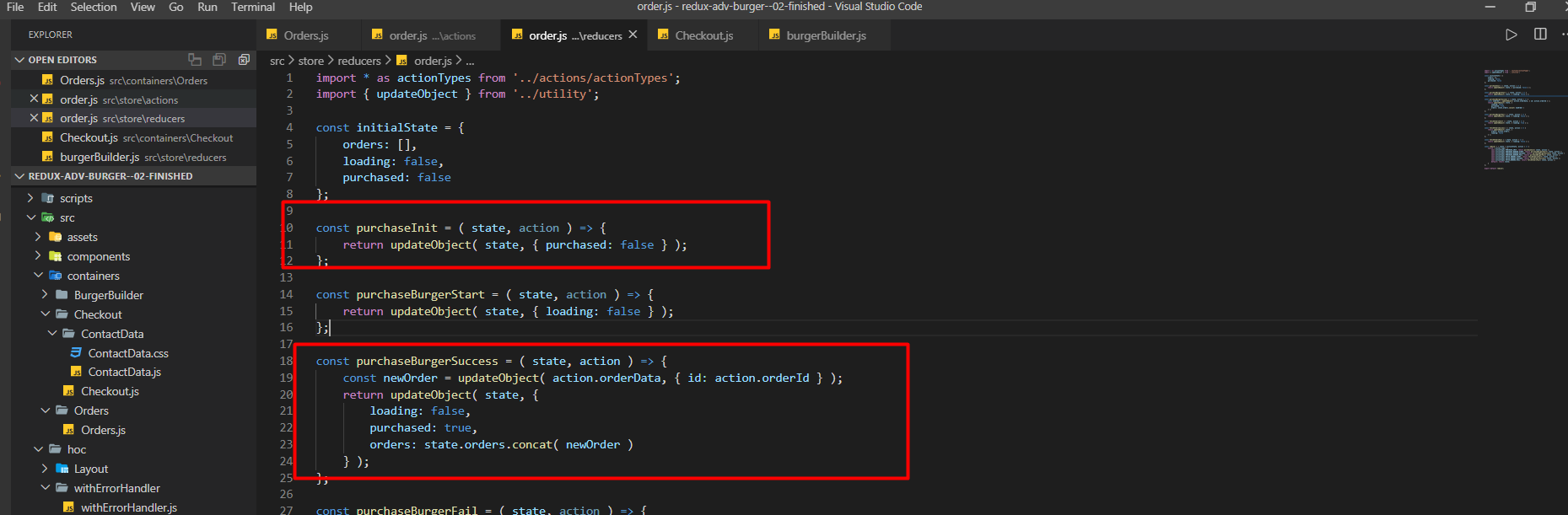


Dùng redux dev tool để kiểm tra các giá trị trong reducer

15. Handling Purchases & Updating the UI

Sau khi mua hàng xong cần set biến purchased true để hiện ra order sumary sau đó mua xong cần redirect trở về giao diện chọn item (root) và set biến purchased trở về false để order sumary ko hiện nữa

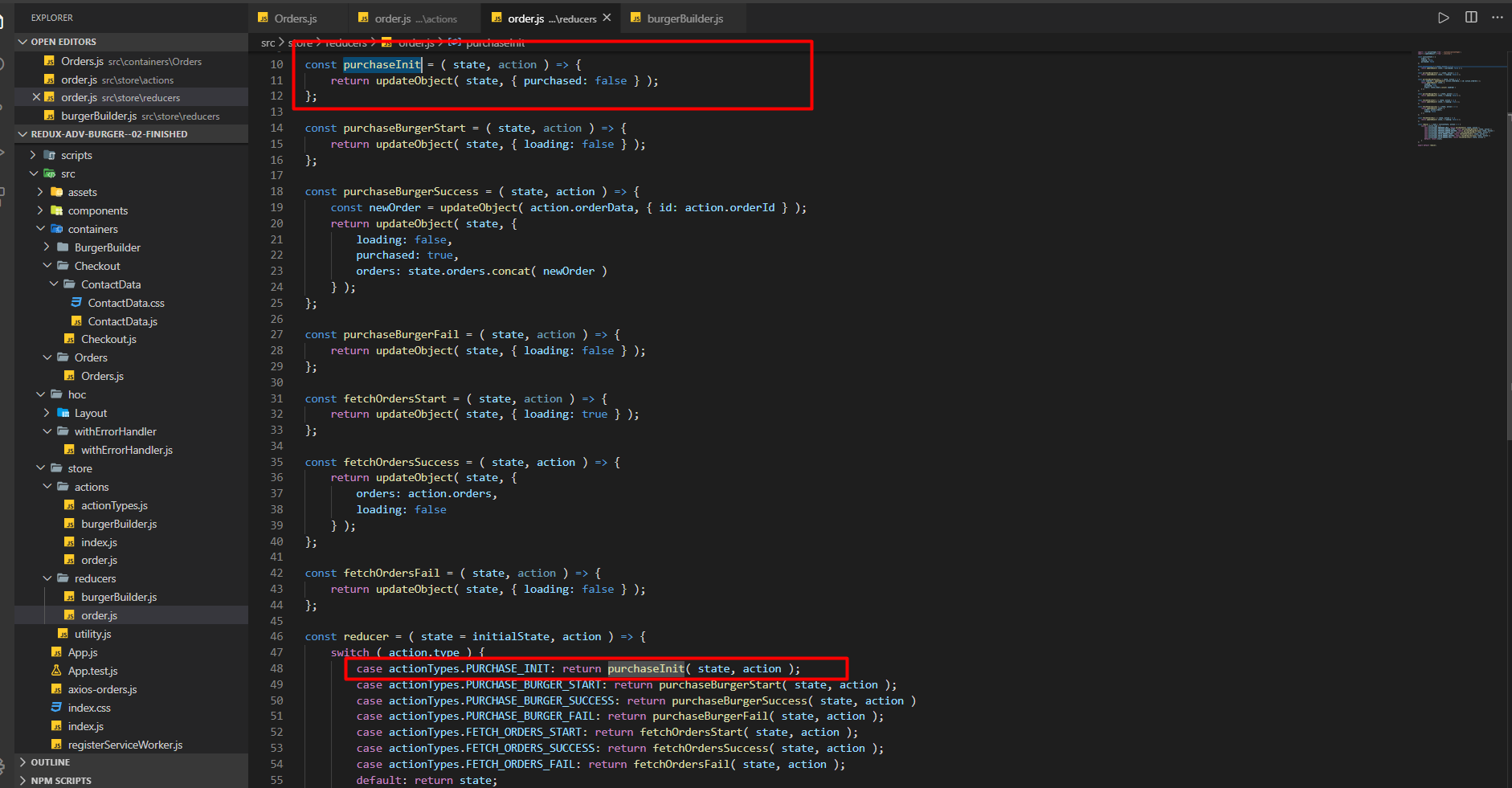
Có thể đưa vào trong action của post order, sau khi post xong thì set biến purchase, và trên hàm check out check biến purchase để redirect: vừa chạy async set biến puchased = false, chạy xong set = true, chạy lỗi set bằng false



Hoặc tạo action mới là purchase init

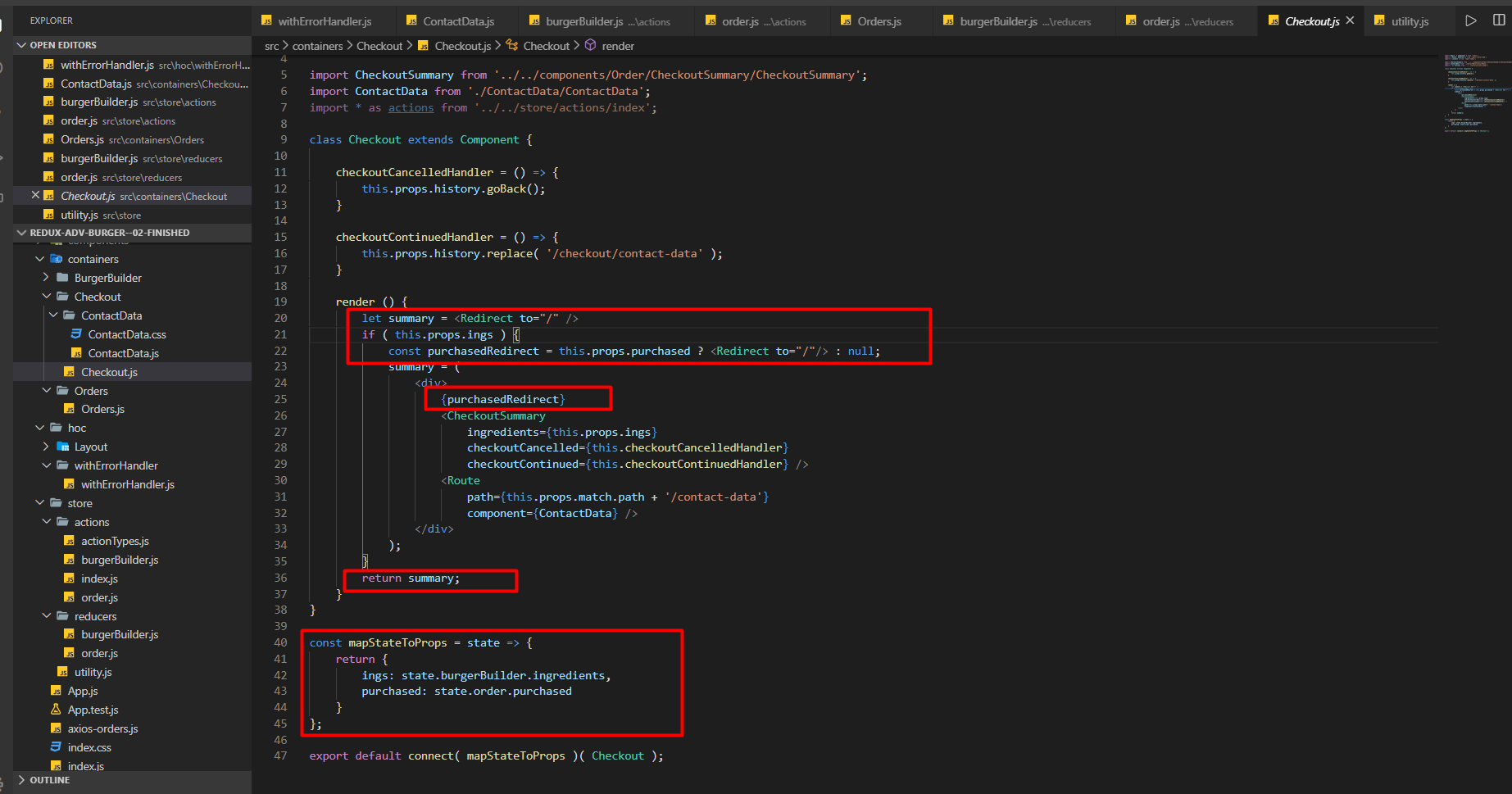
Khi init thif purchased = false

Success thì purchased = true



Nhìn lên hàm success sẽ set purchased true hì, vì order được rồi thì set true để hiện cái order sumarry lên





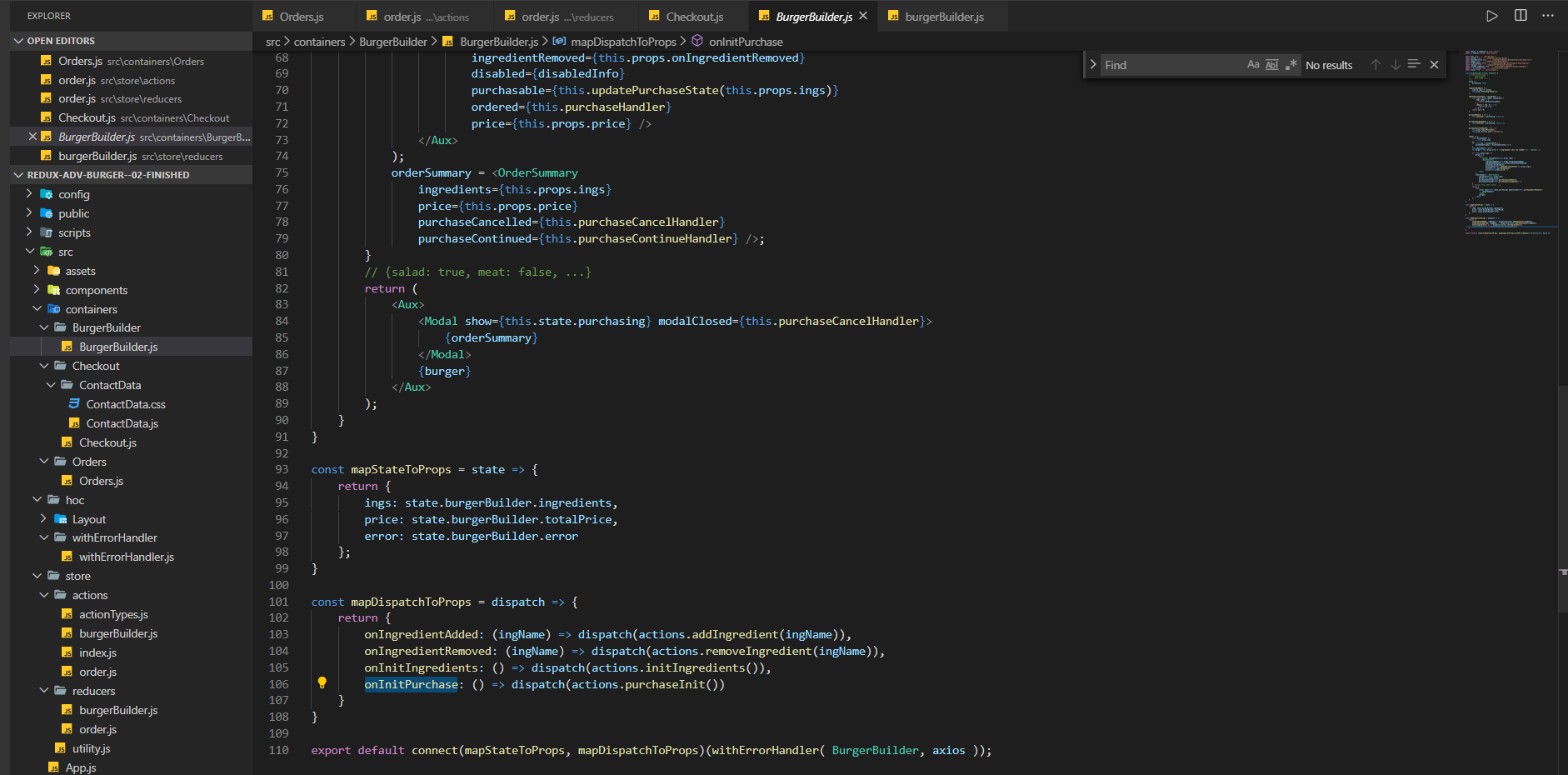


Nghĩa là khi mua thành công thì purchasedRedirect sẽ redirect về root /

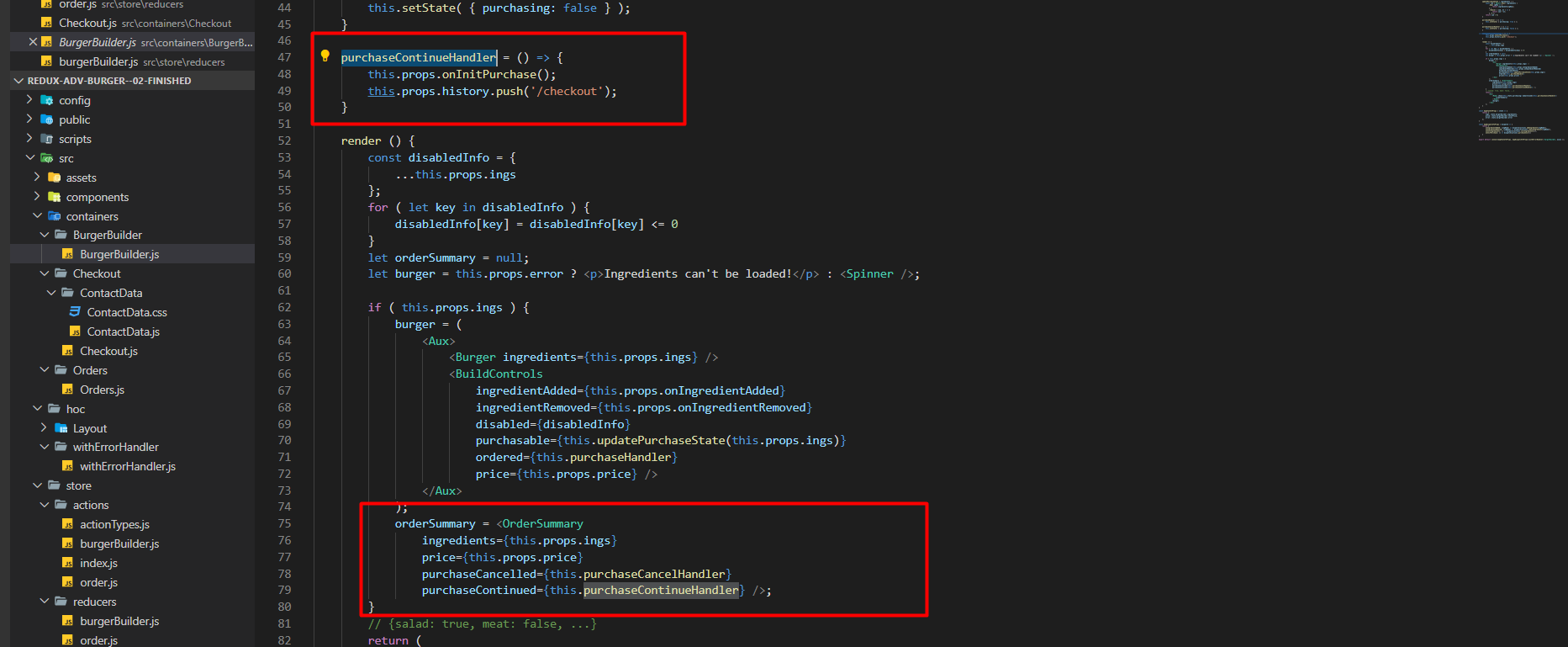
Tuy nhiên làm như vậy, ở lần thứ 2, checkout chưa render ra nên chưa có willmount, nhưng khi render ra thì willmount chưa prevent về false được mà nó vẫn là true -> vẫn redirect

Check out là component điền thông tin người mua rồi post lên

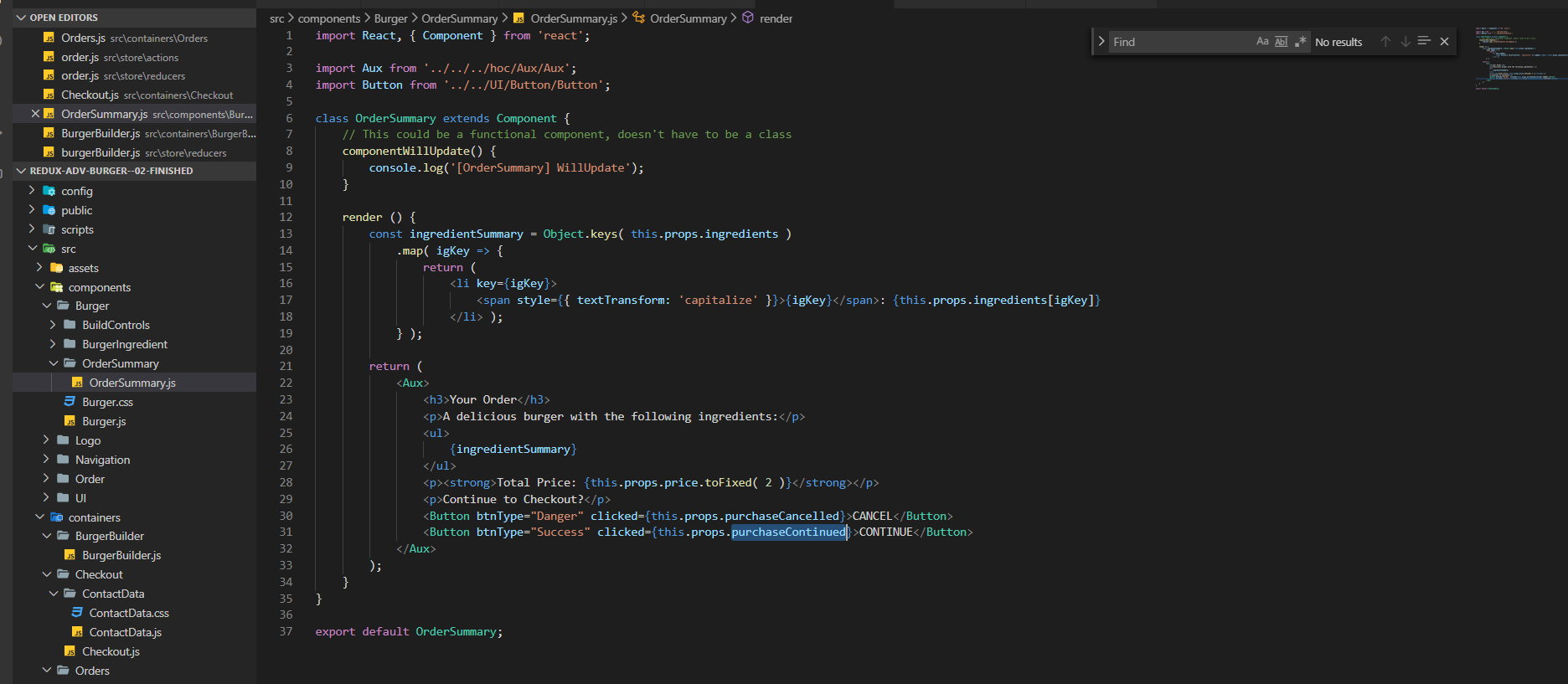
Chưa render ra nên chưa map props với state được



Thay vì đó handler purchased bằng redux trên Burger builder

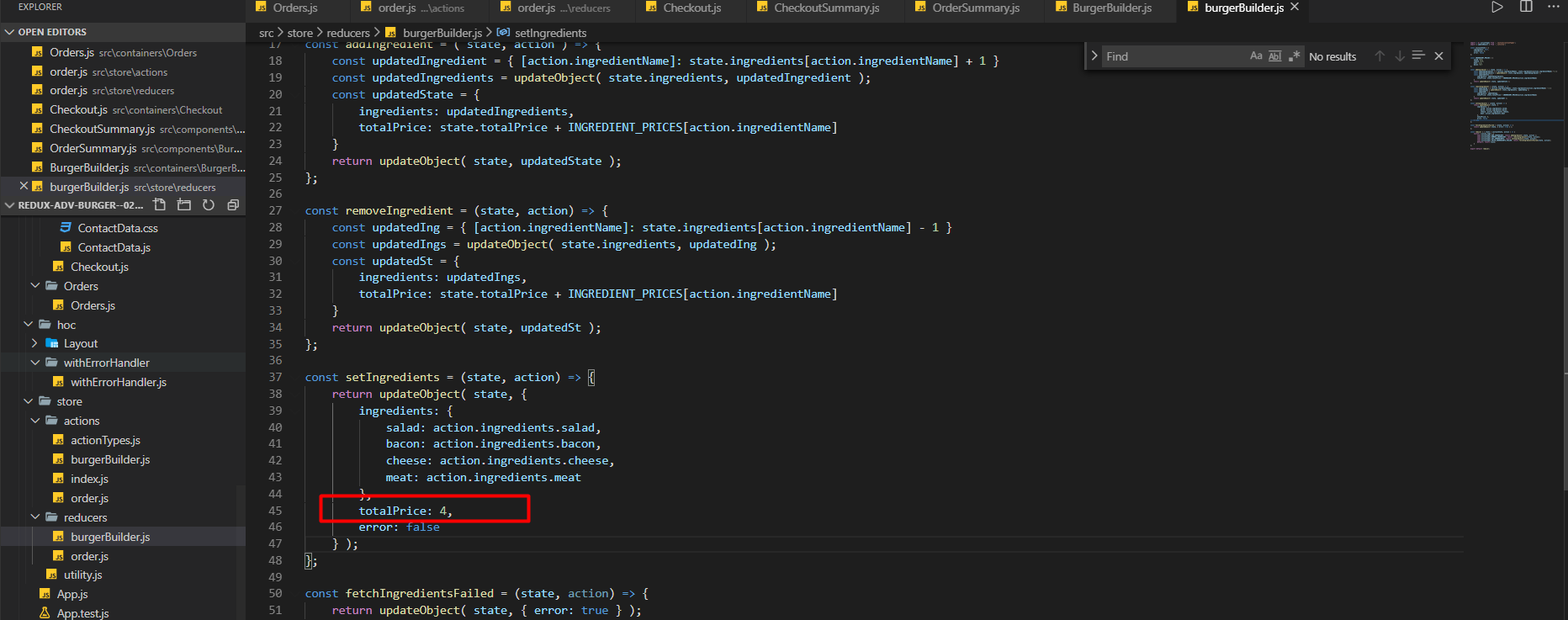


Gọi và truyền qua hàm đó qua orderSumary bằng props (order sumary là component hiện lên khi chọn xong rồi bấm order, hiện ra tóm tắt các item đã chọn)



Và thực hiện đưa biến purchase về true false rồi redirect sau khi nhấn nút nhấn

16. Resetting the Price after Purchases

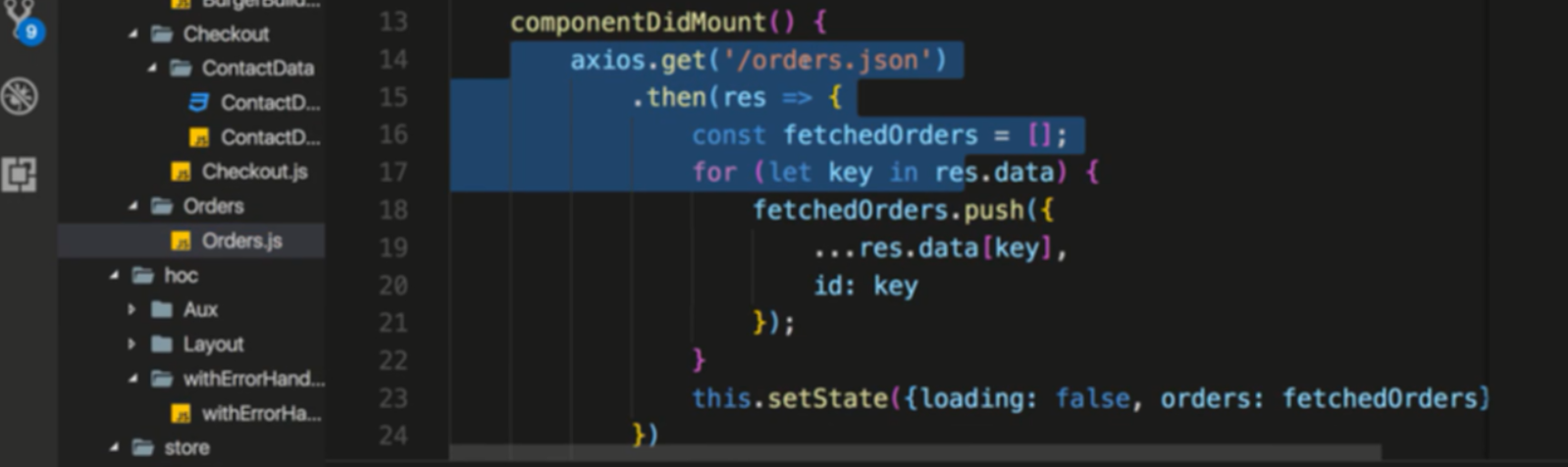


Khi set ingredient thì phải set thêm price ban đầu, nếu không thì price sẽ không được set lại mà lấy price cũ

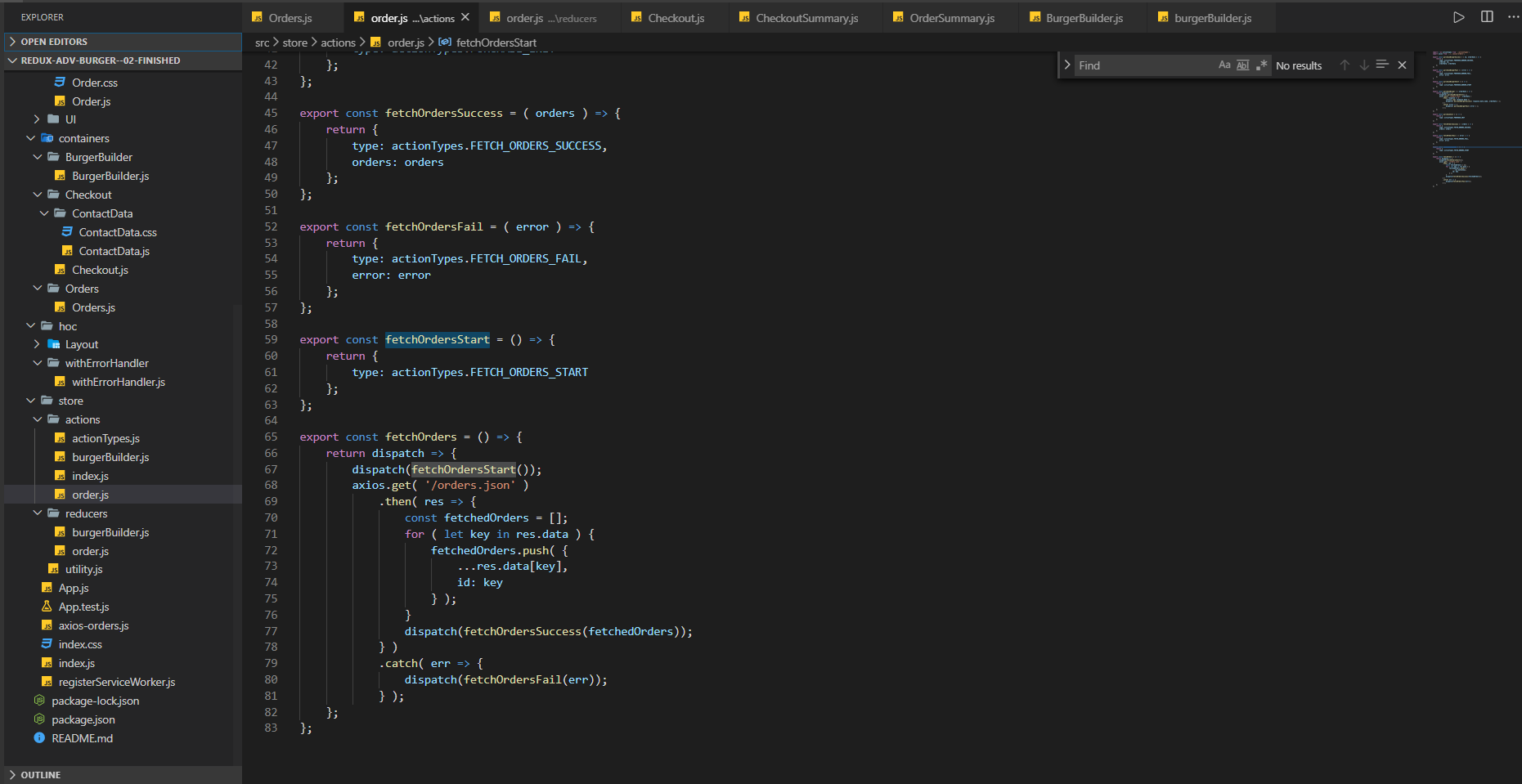
17. Fetching Orders (via Redux)

Trang order sẽ hiện những order đã được order

Giờ thực hiện lấy những order đã được order bằng redux

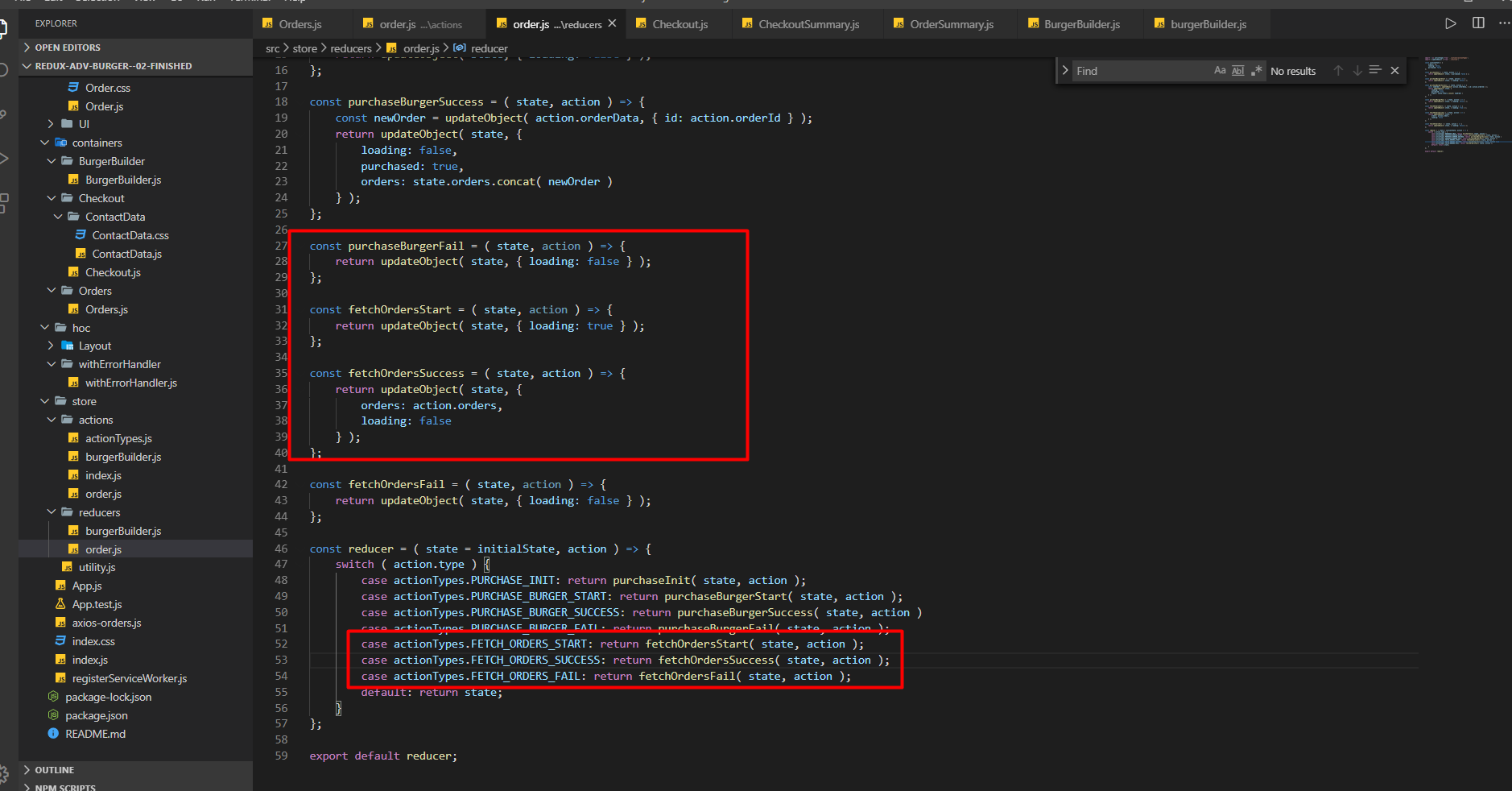


Thay thế đoạn didmount này dùng redux



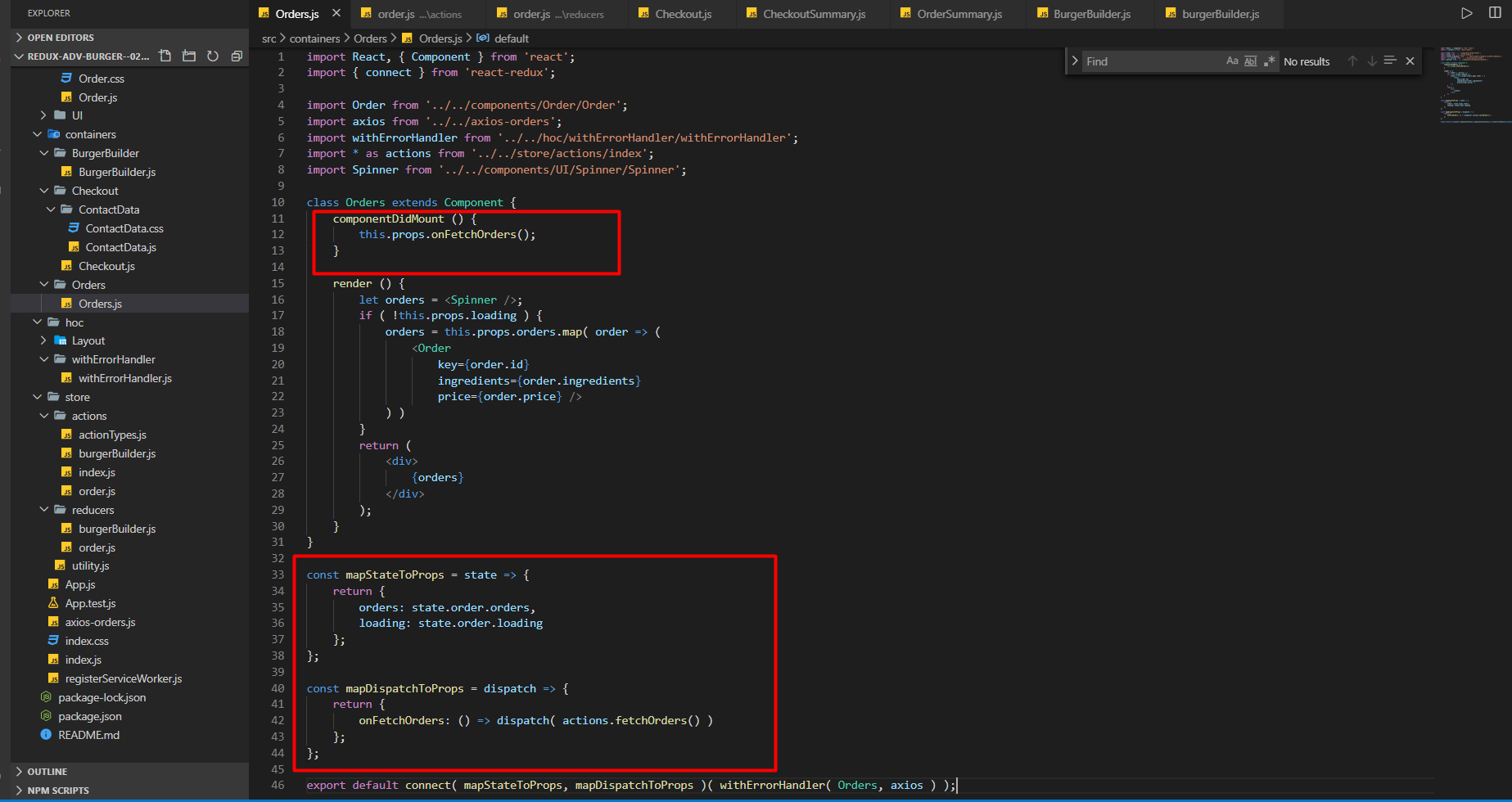
Action, chủ yếu là action fetchOrderStart, những action khác hổ trợ cho action này

Change data theo kiểu muốn lưu trên action luôn, rồi đưa vào reducer chỉ lưu thôi



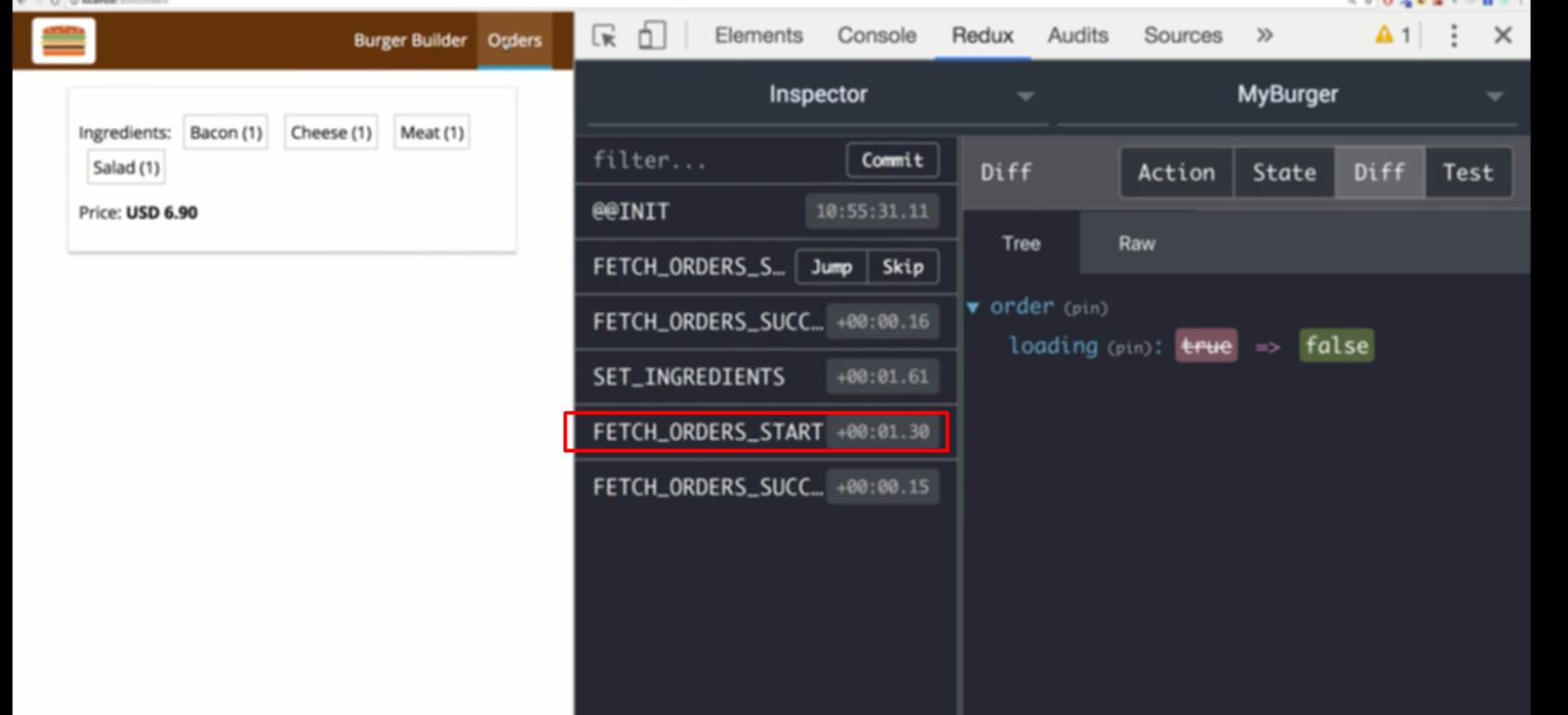
Reducer, tương tự dùng ultiliti và hàm để reducer rõ ràng hơn

Một số biến có thể thêm vào tại reducer nếu biến đó không liên quan khi truyền action, ví dụ biến loading ở đây

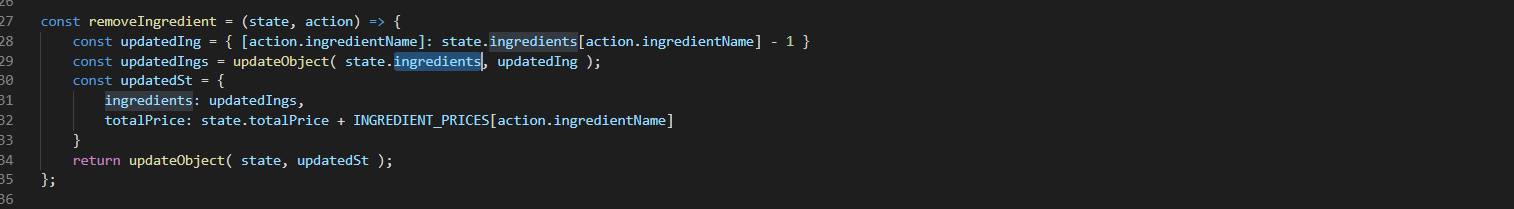


Sử dụng trên order

Thêm spiner vào, khi biến loading true thì hiện cái spiner thay cho nội dung trang



Muốn xem spiner thì bấm jump về đây để load lại



Thấy đoạn này khá hay khi dùng updateObject 2 lần

Lần 1 update 1 thành phần

Lần 2 update toàn object

