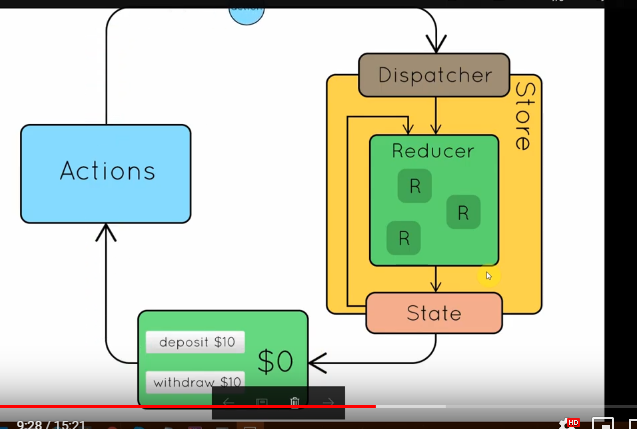
Redux : quản lý các states của các component , khi nào cần sẽ lấy ra



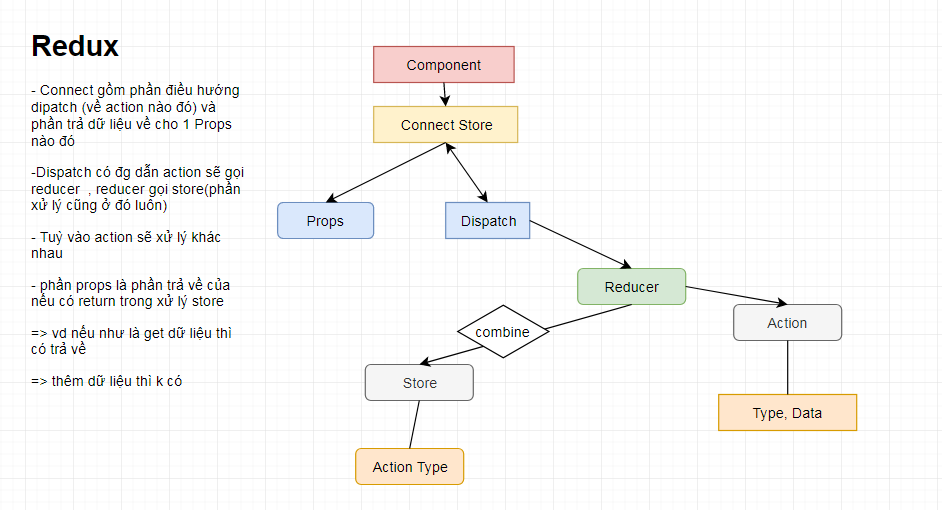
* Ng dung sử dụng các actions sẽ đươc gửi lên reducer phân tích và trả về store các state mới

Điểm mạnh:

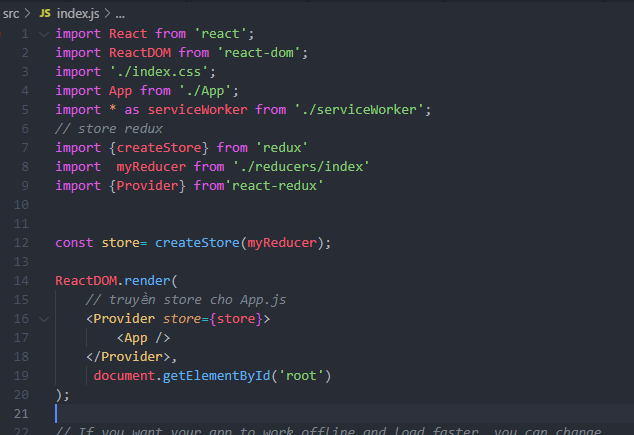
* quản lý dc state , là 1 bản replay có thể undo hoặc redo states
* Tăng tốc phát triễn
* Có ứng dụng offline hay : các thao tác của khách hàng lúc offline sẽ dc ghi nhận và lưu lại , online trở lại sẽ dc gửi lên server

Cài đặt redux : npm install redux react-redux –save (react-redux là để chuyển từ react sang redux)

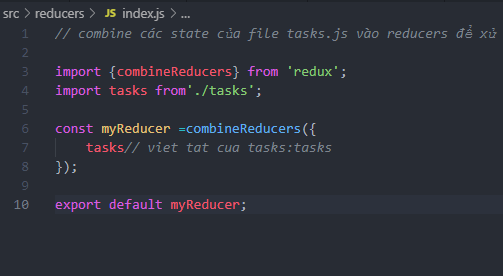
**Xử lý trong redux**



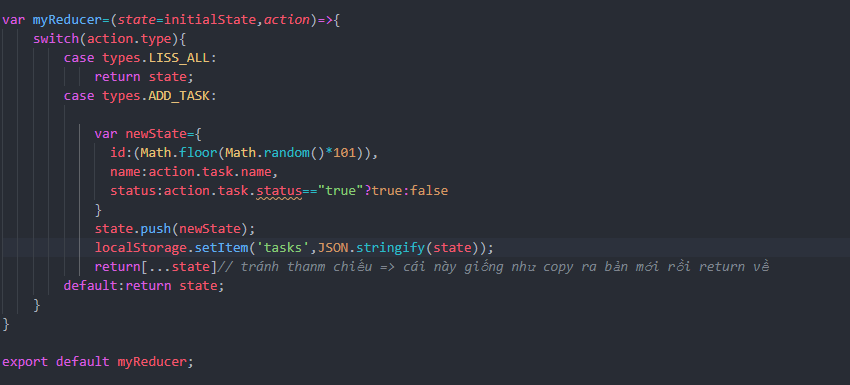
**Connect App với store(index.js)**



**Tạo reducer(./reducer/index.js)**

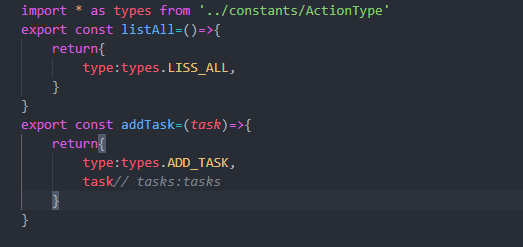


**Tạo store để xử lý(./reducer/tasks.js)**

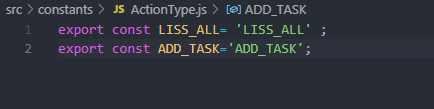
 

**Tạo các action (./action/index.js)**

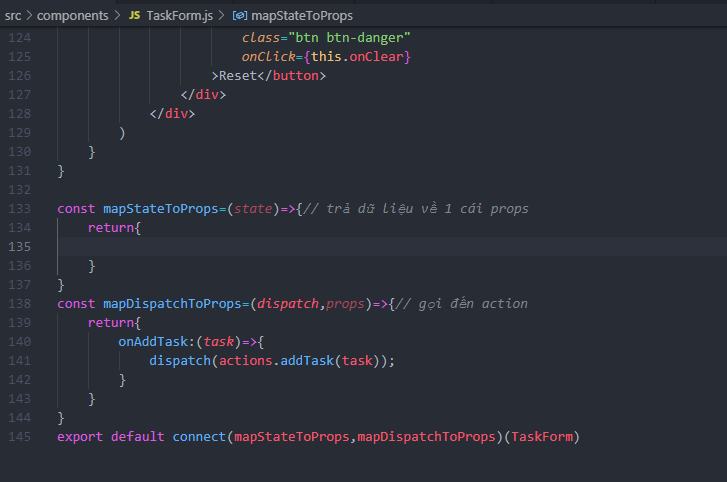
* **Gồm type của action( để kiểm tra vào store xử lý theo action)**
* **Và đầu vào input của action đó (vd tìm kiếm thì có id)**



**Các Action Type(./constant/ActionType.js)**



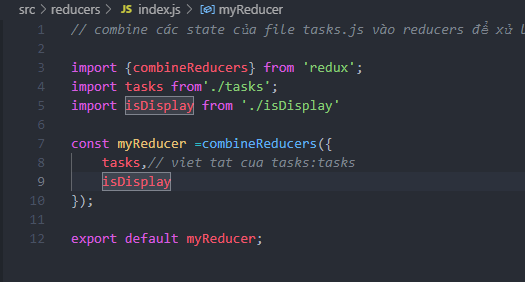
**Connect Component với Store**

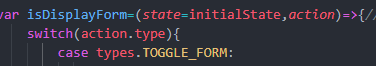


mapStateToProps là Props (phần trả về)

mapDispatchToProps là Dipatch chuyển hướng action

1 reducer sẽ quản lý tất cả các store mình cần



* Lúc gọi đến dispatch thì k quan tâm store nào hết chỉ cần quan tâm là action type là gì
* Reducer sẽ tự check action của store nào và xử lý (dispatch chỉ cần quan tâm action)
* Phần check action của store nào

Sau khi gán state = initialState thì mỗi lần gọi 1 action state thay đổi tiếp tục chứ k phải là mỗi lần gọi là lại lấy lại initialState

* Như api , sau khi fetch products từ api và gọi action gán nó vô state=> cũng y như initialState ,nó trở thành state , và cứ gọi action xoá , hay sửa thì state sẽ thay đổi