# Bước 1: Khai báo state trong Reducer

Trong reducer sẽ bao gồm state và các xử lý logic biến state

Tạo file /redux/reducerList.js

// reducerList.js

const initialState = ['Item 01']

const reducerList = (state = initialState, action) => {

return state

}

export default reducerList

Tiếp tục tạo file /redux/reducerTodo.js

// reducerTodo.js

const initialState = ['thong20']

const reducerUser = (state = initialState, action) => {

return state

}

export default reducerUser

# Bước 2: Khai báo Store và import state vào store

Tạo file /redux/store.js

Trong hướng dẫn này, ta sẽ sử dụng không combineReducer()

Không sử dụng combineReducer:

// store.js

import {createStore, combineReducers} from 'redux'

import reducerList from './reducerList'

import reducerUser from './reducerTodo'

const store = createStore(reducerList, reducerUser)

export default store

Sử dụng combineReducer:

// store.js

import {combineReducers, createStore} from 'redux'

import reducerUser from './reducerUser'

import reducerList from './reducerList'

const reducer = combineReducers({

todo: reducerUser,

list: reducerList

})

const store = createStore( reducer )

export default store

# Bước 3: render state

Để trích xuất được state trong Redux, ta có thể sử dụng hàm connect() để truyền state Redux vào props của component

Hàm connect(arg1, arg2) nhận vào 1 hoặc 2 tham số

Với arg1, bắt buộc phải là 1 function, và function này nhận và 1 tham số là state của Redux

Trong file /screens/FormScreen.js

// FormScreen.js

import React, { useState } from "react";

import { connect } from 'react-redux'

import { Button, Form, FormGroup, Input,

Container, Row, Col,

} from "reactstrap";

function FormScreen(props) {

const {listItem} = props

const [text, setText] = useState("");

return (

<Container>

<h1>Redux & react-redux</h1>

<Row>

<Form>

<FormGroup row >

<Col />

<Col sm={6}>

<Input

type="text"

name="todo"

id="todo"

placeholder="Type todo"

onChange={(str) => setText(str.target.value)}

/>

</Col>

<Col>

<Button color="primary" onClick={() => {}}>Submit</Button>

</Col>

<Col />

</FormGroup>

</Form>

</Row>

{

listItem.map((item, idx) => (

<Row key={`key-${idx}`}><p>{item}</p></Row>

))

}

</Container>

);

}

const mapStateToProps = (state) => {

return ({

listItem: state

})

}

export default connect(mapStateToProps)(FormScreen)

# Bước 3: Khai báo action/dispatch

Dispatch - có thể hiểu 'dispatch' như là 1 phiếu lệnh phải thi hành, bên trong nó sẽ ghi chi tiết những công việc phải thực thi

Ví dụ: trong thực tế, ta có 1 phiếu lệnh (dispatch) nhập kho như sau:

- Tên phiếu: Phiếu nhập

- Sản phẩm: ổ cứng SSD

Còn trong redux, nó là 1 object có 2 key: type và payload

Ví dụ:

{

type: 'ADD\_ITEM',

payload: 'Ổ cứng SSD'

}

Trở lại, ta sẽ định nghĩa các **actions/dispatch**, các actions phải là 1 function

Vd: khi ta click button "Add Item" thì sẽ gọi hàm action, và truyền Item vào action để xử lý, và phải return về 1 object có 2 key: type và payload

Vd: trong file /redux/actions.js

// actions.js

function actionAddItem (str){

return ({

type: 'ADD\_ITEM',

payload: str

})

}

export { actionAddItem }

# Bước 4: Xử lý button "Add Item"

Đầu tiên ta phải import action => import action này vào hàm connect() ở tham số thứ 2, vì tham số thứ 1 là state của Redux, đây là bắt buộc

Sau bước import vào connect(), thì bên trong props sẽ có 1 actionAddItem, ta chỉ việc móc ra để sử dụng bên trong component FormScreen

**\*\*\* Chú ý cú pháp import tham số thứ 2 của ở connect()**

connect(mapStateToProps, {actionAddItem})(Form)

với: cú pháp {actionAddItem} chính là Object destructuring => vậy nó tương đương {actionAddItem: actionAddItem}

vậy actionAddItem sau dấu 2 chấm chính là 1 func, còn trước dấu 2 chấm là key

vậy tham số thứ 2 của connect() ở đây là 1 object

=> <Form actionAddItem={actionAddItem} listItem={state}> … </Form>

Ở file /screen/FormScreen.js

// FormScreen.js

import React, { useState } from "react";

import { connect } from 'react-redux'

import { actionAddItem } from "./redux/actionAddItem";

...

function FormScreen(props) {

const {listItem, actionAddItem} = props

const [text, setText] = useState("");

const addItem = (str) => {

actionAddItem(str)

}

return (

...

<Col>

<Button color="primary" onClick={() => {

addItem(text)

setText('')

}}>Submit</Button>

</Col>

...

}

const mapStateToProps = (state) => {

...

}

export default connect(mapStateToProps, {actionAddItem})(FormScreen)

# Bước 5: xử lý logic cho state trong reducerList

Khi ta nhấn nút "Add Item" thì nó sẽ gửi giá trị của ô input tới Redux xử lý với dispatch là:

{

type: 'ADD\_ITEM',

payload: 'item 02'

}

Trong file /redux/reducerList.js

// reducerList.js

const initialState = ['thong20']

const reducerList = (state = initialState, action) => {

switch (action.type) {

case 'ADD\_ITEM': {

state.push(action.payload)

return state

}

default: return state

}

}

export default reducerList

Ở đây action có 2 thuộc tính mặc định: type và payload

Với:

action.type là giá trị của type trong file /redux/action.js

action.payload là giá trị của payload trong file /redux/action.js

Vậy là xong Redux cơ bản,