**Redux** là 1 thư viện giúp chúng ta quản lý các **state** 1 cách tốt hơn. Thay vì phải truyền state qua từng Component thì **Redux** sẽ tạo ra **1 store duy nhất** dùng để thay đổi dữ liệu.

Redux gồm có 4 thành phần chính là:

1. Store: Là 1 object lưu trữ state của toàn bộ ứng dụng. cho phép truy cập State qua getState(), update State qua dispatch(action).

import {createStore} from 'redux';

import rootReducers from './reducers';

const store = createStore(rootReducers);

* **Action Creators**: Là nơi tạo ra các action dùng để mô tả event do người dùng tạo ra.

export const selectPost = (post) => {

return{

type: "SELECT\_POST",

payload: post

}

}

* **Reducer**: Là 1 fuction nhận đầu vào là state và các mô tả về event và dựa trên đó để trả về state tiếp theo nhưng ko thay đổi state cũ

export default function activePostReducer (state = initialState, action) {

switch (action.type){

case "SELECT\_POST":

return action.payload;

default:

return state;

}

}

Nguyeemn lí hoạt động của Redux

* **Bước 1**: Khi có 1 sự kiện (event) như là GET, POST, UPDATE, DELETE... thì thằng action creators sẽ sinh ra 1 action mô tả những gì đang xảy ra.
* **Bước 2**: Action sẽ thực hiện điều phối Reducer xử lý event thông qua hàm dispatch(action).
* **Bước 3**: Reducer dựa vào những mô tả của Action để biết cần thực hiện thay đổi gì trên State và thực hiện update.
* **Bước 4**: Khi State được update thì các trigger đang theo dõi state đó sẽ nhận được thông tin update và tiến hành render lại phần view để hiển thị ra cho người dùng

**3 nguyên tắc trong Redux**

Redux được xây dựng dựa trên 3 nguyên lý:

* **Store** luôn là nguồn dữ liệu đúng và tin cậy duy nhất.
* **State** chỉ được phép đọc, cách duy nhất để thay đổi **State** là phát sinh một Action, và để Reducer thay đổi State.
* Các fuction Reducer phải là **Pure function** (với cùng 1 đầu vào chỉ cho ra 1 đầu ra duy nhất)