FLUX là một khái niệm trong lập trình giúp ta có thể quản lý flow đường đi của dữ liệu từ component này sang component khác theo 1 hướng duy nhất.

Ví dụ: ta có component cha trong component cha ta có các component con, trong con thì lại có thêm các component cháu. Dữ liệu sẽ được truyền từ cha tới con, rồi con truyền cho cháu, 1 đường đi từ trên xuống dưới. giúp ta xát dịch và quản lý code được tốt hơn.

FLUX gồm 4 thành phần:

* Action: Là nơi dispatch các sự kiện
* Dispatcher: Là trái tim của ứng dụng. Tất cả dữ liệu được dispatch và gửi xuống các stores.
* Store: Nơi chúng ta lưu trữ, cập nhật state dữ liệu.
* View: Dùng để nhận data từ store và render ra giao diện.
* **The Action Creator**

The action Creator có nhiệm vụ tạo ra các action là bước đầu để thay đổi và tương tác đều đi qua. Mọi trạng thái của web app hay render của view có sự thay đổi thì đây là hành động sẽ được tạo ra. Nó sẽ biết rõ bạn cần truyền đạt gì và thực hiện đánh ra văn bản dựa theo định dạng nào đó để cho mọi người đều hiểu được. Việc định nghĩa trước các type của action thành hằng số có sẵn là rất tốt bởi vì khi dev mới nhảy vào trong team thì sẽ mở được file định nghĩa ra. Và chỉ cần nhìn qua một lượt tất cả các type của action sẽ vẫn luôn tồn tại.

* **The Dispatcher**

The Dispatcher được hiểu cơ bản là tập hợp các callbacks, nó sẽ luôn biết trước các danh sách store để gửi action đến. Khi action được gửi đến với dispatcher thì nó sẽ đến với store tương ứng dựa theo quy tắc đồng bộ. Flux dispatcher thường khác với dispatcher ở nhiều kiến trúc khác nhau và nó được gửi đến các store để đăng ký với dispatcher không kể type là gì. Mỗi một store sẽ nhận “nghe” từ các action và tự filter để xử lý chúng.

* **The store**

The store có nhiệm vụ giữ toàn bộ các trạng thái và logic để chuyển đổi trạng thái của app. Tất cả mọi thay đổi liên quan đến trạng thái đều sẽ được thực thi trực tiếp ở đây. Nếu như bạn muốn thay đổi trạng thái cho chúng thì cần phải tạo ra action, submit vào the action creator đi qua với the dispatcher rồi mới được the store xử lý chúng. Bởi vậy, một store sẽ nhận được rất nhiều action và trong store thông thường sẽ sở hữu một cấu trúc switch để quyết định xem có cần quan tâm đến action hay là không.

REDUX:

* Là thư viện js dùng để quản lý và cập nhật state của ứng dụng
* Là một khuôn mẫu

QUẢN LÝ GLOBAL STATE:

* Các component tại mọi nơi trong ứng dụng có thể truy xuất và cập nhật
* Giải quyết vấn để của React khi muốn truyền dữ liệu vào các cấp con cháu.

KHÁI NIỆM REDUX:

* State managerment
* Chú ý nên dùng Imutation là không được gán trực tiếp mà phải dùng Spread Operator (...obj)
* STORE:

{REDUCER:

- Là 1 function cơ bản nhận vào 2 tham số state: init value và action => Kiểm tra action là gì rồi thực hiện logic cập nhật state tương ứng.

- sử dụng theo Imutation

- Không được có code bất đồng bộ ở reducer, k sử dụng math.random(), date.now()

- Làm cho code có thể dự đoán được giá trị

}

* ACTION: là 1 object thường có {Type, payload}

- Action creators: giúp k phải viết action lặp đi lặp lại.

DISPATCH:

* Là 1 function nhận vào 1 action, rồi bắn cái action vào store