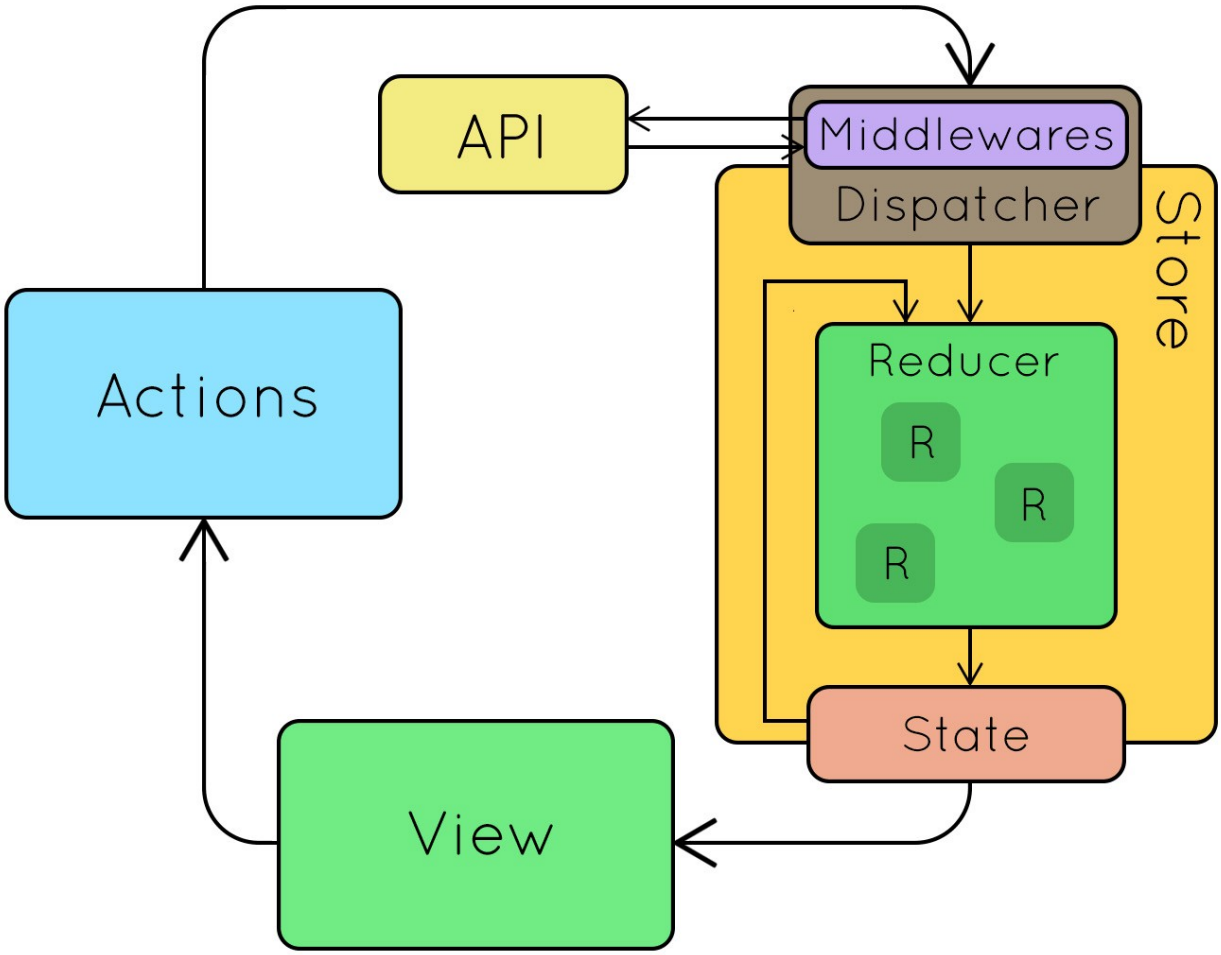
Giới thiệu tổng quan về Redux 🎉

1. Redux là gì? Kiến trúc của nó ra sao?
2. Khi nào cần sử dụng Redux?
3. Redux có phải chỉ để dùng với ReactJS hay không?
4. Các thư viện làm việc với redux

1. Redux là gì? Kiến trúc của nó ra sao?

> Redux is a predictable state container for JavaScript apps.

- Thư viện js quản lý state, mà state này có thể dự đoán được.
- Sử dụng kiến trúc uni-directional data flow.



> https://miro.medium.com/max/2880/1*QERgzuzphdQz4e0fNs1CFQ.gif

- `Store` gồm có:
  - `State`: là dữ liệu hiện tại được lưu trên state.
  - `Reducer`: là hàm biến đổi state cũ sang state mới.
  - Dispatcher: quản lý `middlewares` và chuyển dữ liệu xuống reducer.
- `Action` = plain javascript object mô tả hành động.

2. Khi nào cần sử dụng Redux?

- Dữ liệu được sử dụng ở `nhiều nơi`
- Có hỗ trợ chức năng `undo/redo`
- Cần `cache dữ liệu` để tái sử dụng cho những lần sau.

> Nếu app đang chạy tốt mà không có Redux, vậy có thể app đó không cần tới Redux.

3. Redux có phải chỉ để dùng với ReactJS hay không?

- Không nhé bạn. Nó được sử dụng cho các javascript apps.
- Vậy có thể là:
  - ReactJS
  - Angular
  - VueJS
  - Pure javascript App
  - Other javascript app

4. Các thư viện làm việc với redux

- Redux Logger: https://github.com/evgenyrodionov/redux-logger
- Redux Persist: rt2zz/redux-persist
- Redux Undo: https://github.com/omnidan/redux-undo
- And many more: https://redux.js.org/introduction/ecosystem

📝 Nhớ nè:

- Redux sử dụng kiến trúc 1 chiều: uni-directional data flow
- Redux chỉ dùng 1 `store` duy nhất làm Single Source of Truth
- Redux state là READ-ONLY. Muốn update phải dispatch một `action` (js object)
- Những thay đổi của redux state được thực hiện bởi Pure functions (`reducer`)
- Redux có thể dùng cho các javascript apps, không chỉ riêng gì ReactJS.