

# Redux & Redux Toolkit

## Chuẩn bị gì để học series Redux Toolkit Typescript này

Yêu cầu:

- React hook cơ bản
- Hiểu mutate object là gì
- Biết sơ sơ Typescript
- Cài extension Redux Devtools trên trình duyệt

Sẽ thật tuyệt nếu bạn:

- Đã biết cách hoạt động `useReducer` và `useContext`

## Redux là gì?

Redux là một state container cho Javascript apps.

- Là một single store chứa global state
- Thực hiện event => Dispatch các object action vào store => reducer lắng nghe và trả về state update

Với React thì Redux giúp tạo 1 global state, giúp dễ dàng truyền state xuống các component khác nhau mà không gặp phải vấn đề truyền prop quá nhiều bước (cách giải quyết tương tự `useContext`)

Redux giống như `useReducer` + `useContext`

Bạn có thể dùng Redux với bất kỳ thư viện view nào, nhưng thường thì người ta dùng với React.

Đây là cách viết Redux core

```
const ADD_TODO = 'ADD_TODO'
const TODO_TOGGLED = 'TODO_TOGGLED'

// function tạo action, nó return một plain object
export const addTodo = (text) => ({
  type: ADD_TODO,
  payload: { text, id: nanoid() }
})

export const todoToggled = (id) => ({
  type: TODO_TOGGLED,
  payload: id
})

export const todosReducer = (state = [], action) => {
  switch (action.type) {
    case ADD_TODO:
      return state.concat({
        id: action.payload.id,
```

```
      text: action.payload.text,
      completed: false
    })
  case TODO_TOGGLED:
    return state.map((todo) => {
      if (todo.id !== action.payload.id) return todo

      return {
        ...todo,
        completed: !todo.completed
      }
    })
  default:
    return state
}
```

## Data Flow

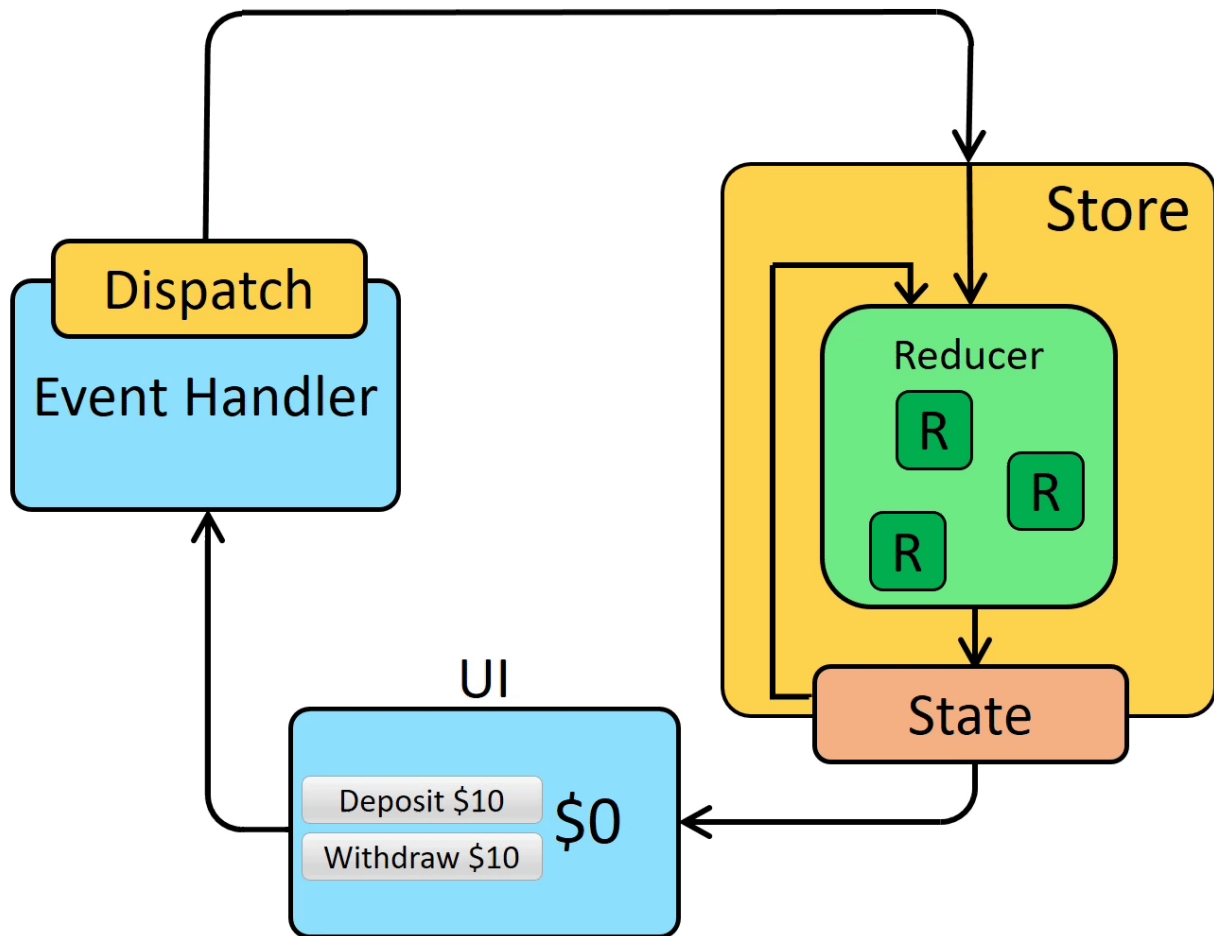
Data flow một chiều của redux có thể tóm gọn như sau

- User click thì sẽ dispatch một action
- reducer lắng nghe các action trong app và xử lý tính toán để trả về một state mới
- component đọc state mới và cập nhật lại UI

### Data flow theo set state thông thường

- User click thì sẽ set state với một giá trị mới
- component đọc state mới và cập nhật lại UI

Flow của set state ngắn hơn so với reducer



Redux sử dụng một vài **thuật ngữ**

- **action**: là một plain object (object đơn giản tạo bằng `{}` hoặc `new Object()`) chứa field là **type** mô tả chuyện gì vừa xảy ra. action là một object nhưng chúng ta thường khai báo là một function return về object để có thể dễ dàng gửi data vào action. Chúng ta thường gọi đó là **payload**
- **reducer**: là một function để tính toán state mới dựa vào previous state + action
- **store**: store sẽ chạy root reducer bất cứ khi nào một action được dispatch
- **dispatch**: là một function dùng để gửi một action đi đến store (hay còn gọi reducer cũng được)

Store là một object chứa state tree. Chỉ nên có duy nhất 1 store trong app.

```
type Store = {
  dispatch: Dispatch
  getState: () => State
  subscribe: (listener: () => void) => () => void
  replaceReducer: (reducer: Reducer) => void
}
```

Có một vấn đề là chỉ để thay đổi một state mà chúng ta tạo ra quá nhiều code thừa: tạo constants, actions, cập nhật state làm sao để không phải mutate state gốc. Và vấn đề này lặp đi lặp lại một cách nhàm chán => Team Redux tạo ra Redux Toolkit để giải quyết vấn đề này

## Redux Toolkit là gì?

Redux Toolkit sinh ra để đơn giản quá cách chúng ta làm việc với redux, tập trung vào logic hơn là những "boilerplate" không cần thiết.

Redux Toolkit mới ra ~ 2018, vậy nên trước đó người ta chỉ dùng Redux thôi. Vì thế một số dự án các bạn join sau này có thể vẫn còn dùng Redux với cách tiếp cận cũ.

Team Redux đều recommend dùng Redux Toolkit trên cả trang <https://redux.js.org/> và <https://react-redux.js.org/>

Bây giờ sẽ có các document sau:

- Document chính thức của Redux: <https://redux.js.org/>
- Document chính thức của Redux cho React: <https://react-redux.js.org/>
- Document chính thức của Redux-Toolkit: <https://redux-toolkit.js.org/>

Chúng ta có thể bỏ qua bước tạo một project bằng redux kiểu cũ, vì đơn giản là nó giống `useReducer` và `useContext`. Học luôn redux toolkit cho tiết kiệm thời gian

Redux Toolkit bao gồm các packages nhỏ sau

- Redux
- ImmerJs: Dùng làm việc với immutable state thuận tiện hơn (cập nhật state dễ dàng)
- RTK query: Fetch & catching API

Cách cài bộ redux toolkit vào app

```
yarn add react-redux @reduxjs/toolkit
```