

1, React Native là gì?

React Native là một framework được tạo bởi Facebook, cho phép developer xây dựng các ứng dụng di động trên cả Android và iOS chỉ với một ngôn ngữ lập trình duy nhất: JavaScript.

Sự ra đời của React Native là lời giải cho bài toán liên quan đến hiệu năng Hybrid và sự phức tạp khi phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động trước đó.

2, Expo là cái gì ?

Expo là một framework để đẩy nhanh việc viết app React Native. Giống như Laravel hay Symphony cho PHP, Ruby on Rail của Ruby. Đồng thời cung cấp một công cụ để chạy thử và debug.

Thực sự Expo rất mạnh, rất nhiều thứ hỗ trợ sẵn cho Android và iOS. Có nghĩa là nếu app đang viết không có gì quá đặc biệt, quá "đỉnh của đỉnh" thì việc sử dụng Expo sẽ mang tới rất nhiều lợi ích.

<https://docs.expo.io/versions/latest/>

3, React Hooks là gì ?

có 2 loại component:

1. stateful component / class component:

```
class Example extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }

  componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
  }

  componentDidUpdate() {
    document.title = `You clicked ${this.state.count} times`;
  }

  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Click me
        </button>
      </div>
    );
  }
}
```

2. stateless component / functional component/

```
import { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Với function component, chúng ta chẳng cần quan tâm tới `super(props)`, `this` chạy thế nào. Chúng ta sẽ bỏ `state`, giải quyết `lifecycle`, chia sẻ logic nữa là xong.

Hook là gì ?

Hooks là những hàm cho phép bạn "kết nối" React state và lifecycle vào các components sử dụng hàm.

Với Hooks bạn có thể sử dụng state và lifecycles mà không cần dùng ES6 class. vd:

1, useState

```
import { useState } from 'react'

function Counter() {
  // Hàm useState() nhận vào một tham số là giá trị ban đầu
  // của state, trả về một cặp [value, handler] chứa giá trị
  // hiện tại của state, và handler để thay đổi state đó.
  const [count, setCount] = useState(0)

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  )
}
```

2, useEffect

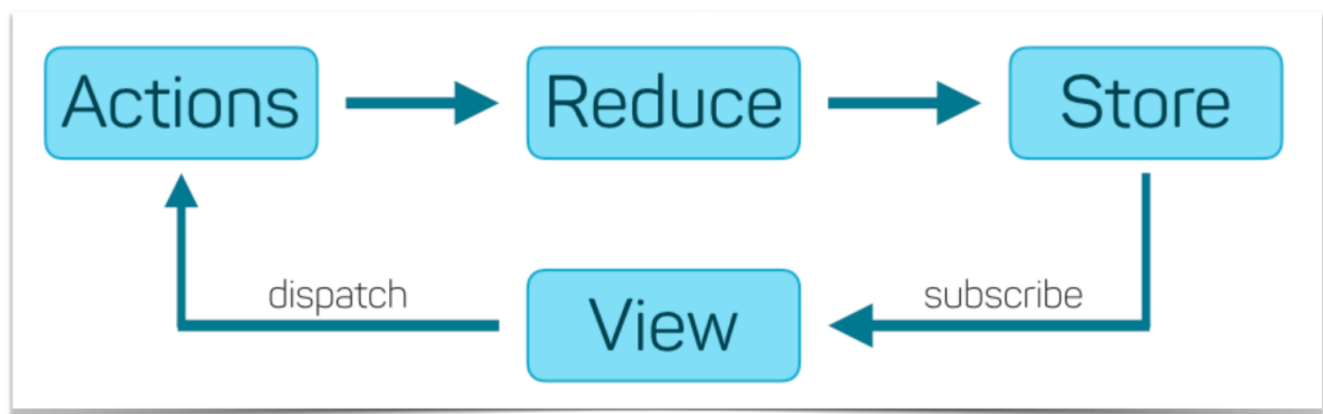
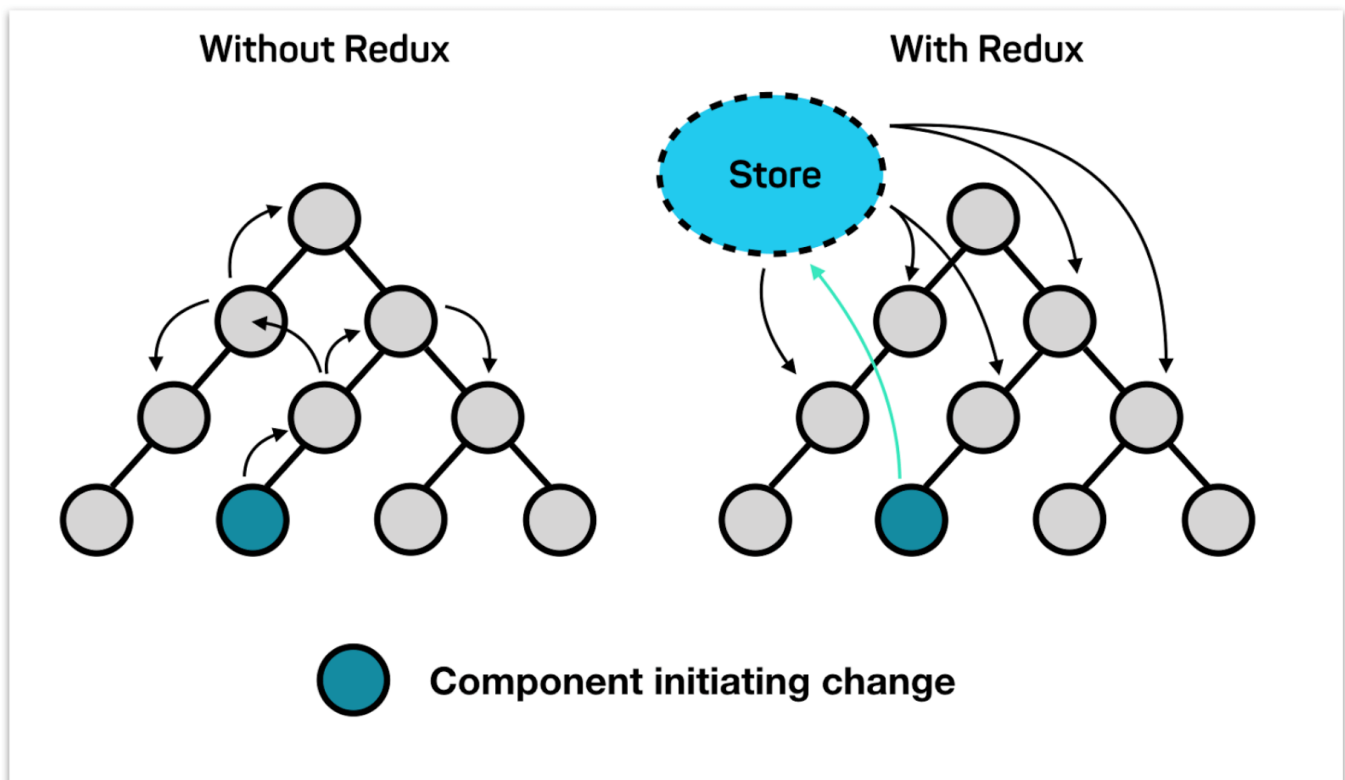
```
import { useState, useEffect } from 'react'

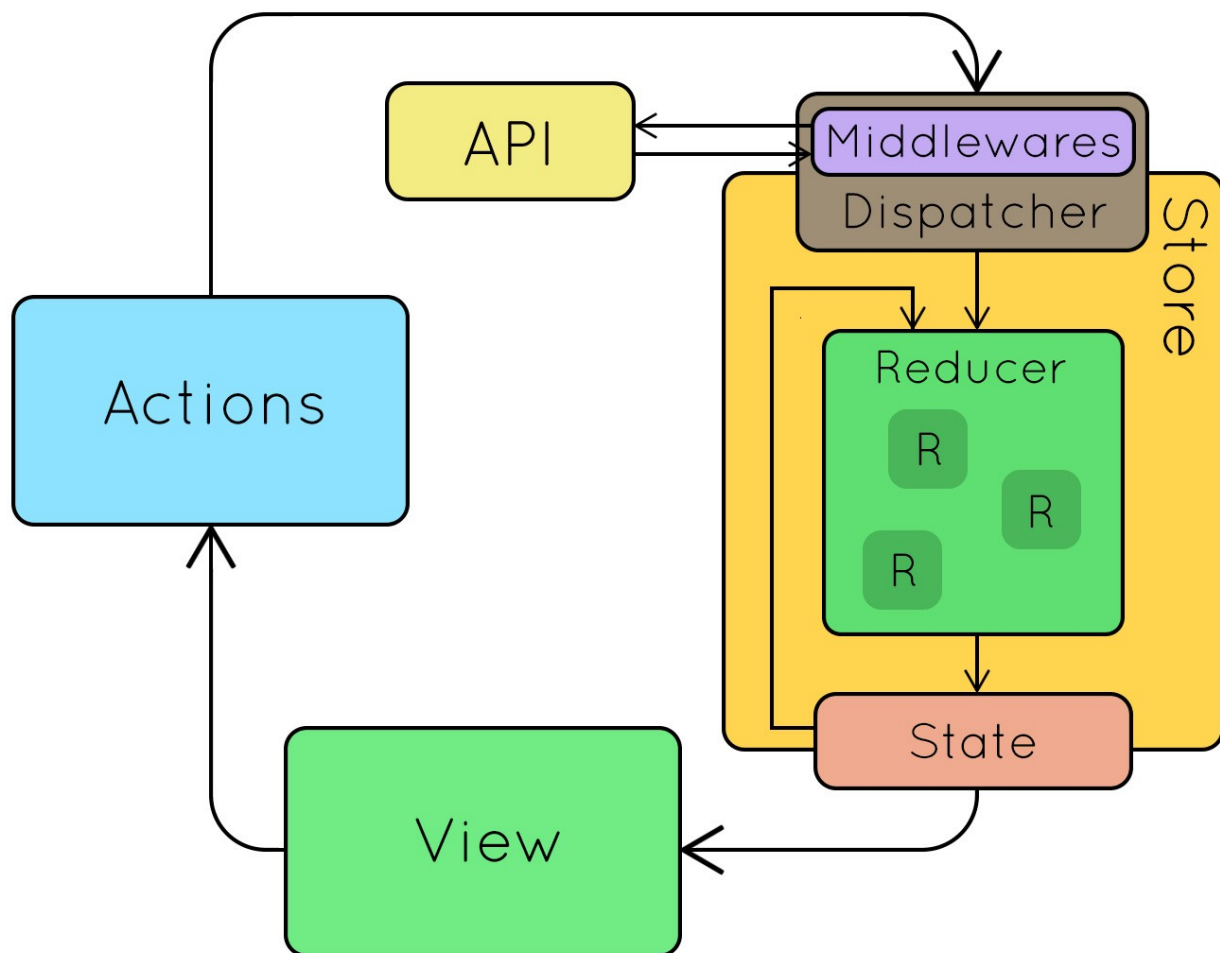
function Example() {
  const [count, setCount] = useState(0)

  // Giống như componentDidMount và componentDidUpdate
  useEffect(() => {
    document.title = `You clicked ${count} times`
  })

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  )
}
```

4, Redux





Cấu trúc 1 RN project

```
|— App.js
|— app.json
|— assets
|— babel.config.js
|— doc.md
|— node_modules
|— package.json
|— package-lock.json
|— README.md
|— src
```

src:

```
|— actions
|— components
|— consts
|— images
|— reducers
|— sagas
```

- └ screens
- └ store
- └ styles