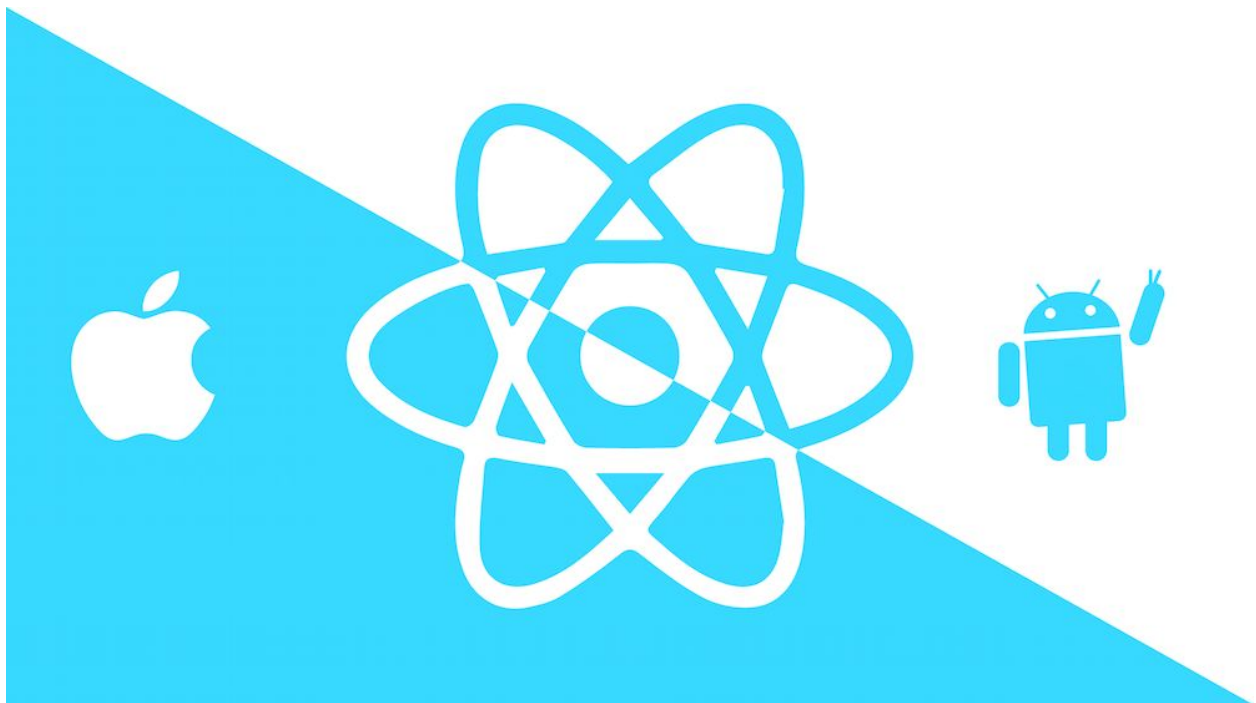


Introduction to React Native

Why, When, What and Hows about what we call RN!



Siavash Jan Ahmadian

Seyed Alireza Moezi

12.28.2020

INTRODUCTION

React Native is an open-source mobile application framework created by Facebook at march 26, 2015. It is used to develop apps for android, iOS, windows, web and etc.

QUICK HISTORY

It all started when using HTML5 for facebook android app resulted in an unstable app the retrieved data slowly. Also the iOS app for facebook could generate UI elements from a JavaScript thread so they decided to perfect the iOS way to build native apps with this technology.

DIFFERENCES BETWEEN REACT DOM AND REACT NATIVE

The main difference is that React Native doesn't manipulate the DOM and runs in background process directly on the end-device and uses serialisation and asynchronous methods to communicate with the native platform. This makes the app work faster and smoother and also the development much easier.

React Native doesn't use HTML or CSS, instead messages from JavaScript threads are used to manipulate native views, also developers could write native code in languages such as Java for android or Swift for iOS which made the development

much more flexible.

REUSABLE UI COMPONENTS?

React Native allows us to create reusable UI components which makes it more declarative and efficient. It lets the developer to build complex UIs from small and isolated pieces of code which are called “components”. Reusable UI components means presenting data that changes over time.

COMPONENTS IN REACT NATIVE

We have two types of components, state-full or class based components and state-less or function based components.

State-full or class based component has its state and data associated with the state. You can pass the data inside this type of component via state or props objects. This type of component is harder to support as they persist some data and may change the global state of the app.

State-less or function based components are just some simple JavaScript functions. They have some limitations and essentially must have one global place to manage state.

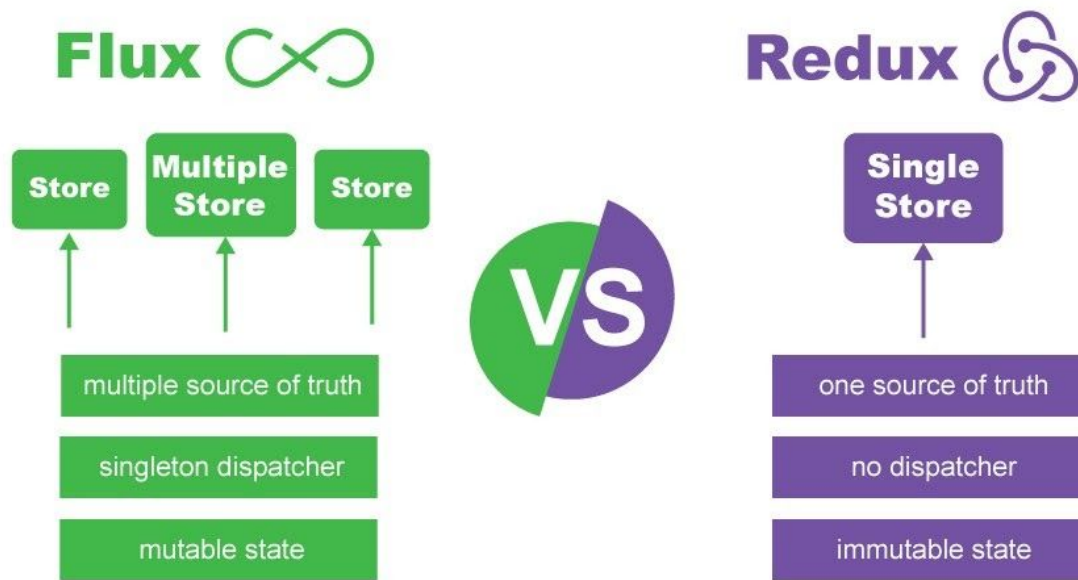
PRESENTATIONAL (HIGH-ORDER) COMPONENTS

High order component or HOC is essentially a design pattern also known as the Decorator pattern. In React Native, HOC is a component which wraps another component by adding extra functionality or extra properties. This allows abstraction from some commonly used logics and keeps the code dry.

CONTAINER COMPONENTS

Container components, on the other hand, have logic to set state or have functions to emit events up to a parent component. The general rule of the thumb is to keep your component as simple as possible with a Single Responsibility design principle in mind, which means the component must do only one thing but do it perfect.

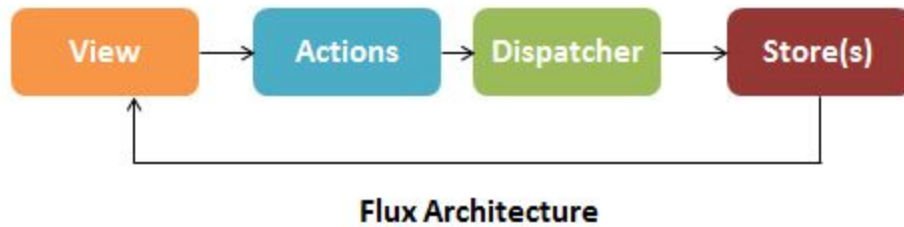
REACT NATIVE ARCHITECTURE: REDUX AND FLUX



FLUX:

After learning about the instability and complexity of the MVC (Model-View-Controller) architecture, Facebook development team released Flux as an alternative to MVC architecture. The Flux architecture is based on the following components:

- store/ stores: serves as a container for the app state and logic.
- Action: enables data passing to the dispatcher.
- View: it's the same as the view in MVC but in the context of React components.
- Dispatcher: coordinates actions and updates to stores.



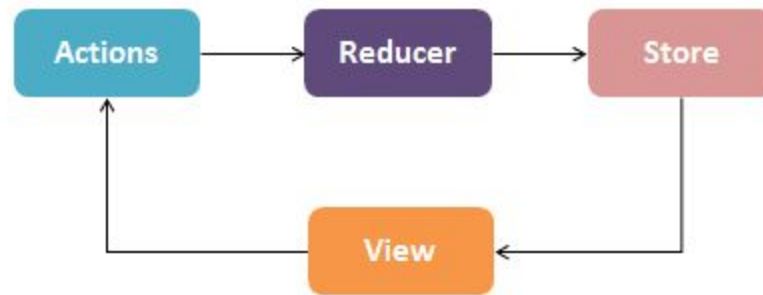
In the Flux architecture, when a user clicks on something, the view creates actions. Action can create new data and send it to the dispatcher. The dispatcher then dispatches the action result to the appropriate store. The store updates the state based on the result and sends an update to the view.

REDUX:

However, Dan Abramov felt that this architecture could be simpler. Consequently, Dan Abramov & Andrew Clark developed Redux in 2015.

Redux is a library, which implements the idea of Flux but in quite a different way. Redux architecture introduces new components like:

- Reducer: logic that decides how your data changes exist in pure functions.
- Centralized store: holds a state object that denotes the state of the entire app.



Redux Architecture

In Redux architecture, application event is denoted as an Action, which is dispatched to the reducer, the pure function. Then reducer updates the centralized store with new data based on the kind of action it receives. Store creates a new state and sends an update to view. At that time, the view was recreated to reflect the update.

REACT JS FEATURES

React JS is a JavaScript library a developer use to construct a high level UI layer. React JS is the heart of React Native and it embodies all React's principles and syntax. React JS has some features that are listed below:

1- JSX: JSX is a JavaScript syntax extension which isn't necessary for development but it is highly recommended. JSX makes components/blocks readable, it displays how components are plugged or combined with.

2- Components: as we talked about them before, React is all about components

and you need to think of everything as a component, this will help you maintain the code when working on larger scale projects.

3- Undirectional data flows and Flux: React implements one-way reactive data flow which makes it easy to reason about your app and reduces boilerplate. Flux is a pattern that helps keeping your data undirectional.

MOBILE DEVELOPMENT: REACT NATIVE

Due to the growing popularity of mobile apps, every company needs a mobile app or apps to stay competitive. One of the main problems is that mostly you have to develop an app for both android and iOS which needs two different skill sets and usually two different development teams in a company. As a result companies have started to adopt cross-platform solutions to build an app for both android and iOS with a single language.

React Native is one of the cross-platform app development frameworks beside Ionic, Xamarin, Flutter and RIBs.

APPS MADE WITH REACT NATIVE

Facebooks, Walmart, Bloomberg, Instagram, Soundcloud, Skype and Tesla are Some of the apps that are made with React Native.