Introduction to React Native

Tong Zhou(tz13)

April 22, 2017

Abstract

This paper is going to talk about some basics of react native, a platform to build mobile application by JS. Then explain its pros and cons and compare it with react and native

1 What is React Native?

React Native has become quite popular among mobile developers. Previously we need to build Android by Java and IOS by Object-C and swift. React Native is a cross-platform technology to build applications for whatever platform they choose, without needing to learn a fundamentally different set of technologies for each.

It takes the advantage of React, which is a library to build web application. The virtual DOM is used for efficient re-rendering of the DOM in React. So, it is very fast to render the elements. Also, because of virtual DOM, react doesn't tightly coupled with DOM, therefore React can render web elements in Web applications and render platform specific elements in different platform such as Android and IOS. Another best thing of react is that it makes app's view layer a pure output of state. Because of separated concern of component and data, react is deterministic, which means every time we dispatch an action to change the state, we can predict what the view will look like. So, this will prone to less errors and make tracing data flow in complicated applications easier.

2 Why React Native becomes popular?

It gives us native experience. it will create real mobile application rather than a "mobile web app" because it will render platform specific components. Most mobile app built with JavaScript use Cordova or a framework built on top of it. With Cordova, developers can make calls to native APIs, but it still just approximate native components by HTML and JavaScript. In React Native, the components we define will end up rendering as native platform widgets. So, it's difficult to distinguish it from apps built by Java or Objective-C.

Also, it's goal is "learn once write anywhere". Developers can just learn JavaScript and react-native platform to develop apps on different platform, which will reduce the cost of development and version control. It gives developers a fast and great development experience, for example, from the FB conference the lecturer said that they only spent 2 weeks to rewrite the Facebook application.

3 Basics of React Native

React is the key feature of React native. They share the same concepts such as Props, component and state.

React Native is component based. Everything in the UI is a component. It includes container component and presentational component. Presentational component will represent view to users. But where does the data come from? Container component will handle state and pass props to them.

Component is static, we will use props and state to customize these components. Props are set by parent and they are fixed throughout the lifetime of a component. We will need state to change the data. But the state is immutable, we just to transfer the old state to a new state, which is easy to trace the data and find out the problem when the view is not shown as what we expect.

How about style? React native uses Flexbox layout model to design the style of mobile application, which solves design issues that are particularly challenging on mobile without requiring CSS or JS tricks. Flexbox is great when we need to accommodate our components and views in different screen sizes or even different devices. All styles are putted into a style prop created by Stylesheet's create method. Then this prop can be applied in components.

4 What's the difference between React Native and React?

Even though React Native is rooted from React, there are many things to learn such as native components.
DOM and Styling is the most obvious difference. React render components by HTML, whereas React Native will map the actual real IOS or Android components. Most components in React Native can have a similar tag in HTML. Such as <Text> in React Native is like <p> in HTML. To style the application, because React Native is not made from web elements so it isn't styled in the same way. Web application would use CSS but React Native would use Flexbox to style the application.
In mobile application development, gestures are more various than in web application. In react native, there's a new way to animate the different components with JavaScript by using Animated API. To interact with user gestures, React-Native provides the PanResponder API to respond to different touch events.

React Native's goal is "learn once, write anywhere". It's different from "write once, run anywhere" because different platform has different specific user interface and features. So, react native provides platform specific code to write specific logic, stylesheet to follow the UI guideline of that platform. But react doesn't have that because it focuses on web development.

5 Cons of React Native

I have discussed about why do I think React Native is great. But it still has disadvantages. It is established from 2015 by Facebook, so it is still a new platform but Android and IOS is mature. Sometimes we don't want to reinvent the wheel, in Android and IOS there are various libraries and resources. But there are not so many 3rd libraries to support React Native and help developers avoid reinventing the wheel. Even though the community is active, the size of this community is far less than that of Android or IOS.
The recycle time is fast in React Native may is because "move fast" culture in Facebook. But it sometimes is a burden to developers because they need to always fix bugs when upgrading the framework.

6 Conclusion

In this paper, I have introduced React Native and compare it with React. Many people say JavaScript and Web is the future. React Native is an example to prove this. This is a young, great and fast-growing platform. Even though it has some cons, we can see many applications have already changed to use React Native to build their whole application or combine some React Native components in their app.
However, I think it's a long way for React Native to take the place of Android and IOS. Nowadays If we want to build a simple page, short-time application, I will recommend React because it's fast to develop and is easy. But if we would like to build a complicated, long-term application, I still recommend to use Android and IOS, because they are mature platform, are more stable than React Native.

7 References

- React Native Layout System
  https://www.overleaf.com/contact
- Why You Should Consider React Native For Your Mobile App
  https://www.smashingmagazine.com/2016/04/consider-react-native-mobile-app/

- What are the main differences between ReactJS and React-Native?
  https://medium.com/@alexmngn/from-reactjs-to-react-native-what-are-the-main-differences-betw

- React Native: Is it the end of native development?
  https://www.novoda.com/blog/is-it-the-end-of-native-development/

- React Native
  https://facebook.github.io/react-native/