

УДК 004.4

Е.В. Зотов

Магистрант 2-го года обучения

МГТУ «СТАНКИН»

Е.С. Занина

Магистрант 2-го года обучения

МГТУ «СТАНКИН»

г. Москва, РФ

ВЫБОР И ОБОСНОВАНИЕ ИСПОЛЬЗОВАНИЯ СОВРЕМЕННЫХ СРЕДСТВ ДЛЯ РАЗРАБОТКИ МУЛЬТИПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Аннотация

В статье рассматриваются современные методы разработки мультиплатформенных приложений с использованием одного ЯП и общей кодовой базой. Данный метод позволяет уменьшить цену разработки мобильных приложений общего назначения и обеспечивает простоту поддержки нескольких платформ.

Ключевые слова:

Мобильные приложения, разработка ПО, мультиплатформенные приложения, react, react native, expo, react js, IOS, Android, Javascript, JSX, Virtual DOM, React Native App, Java, Objective-C

Для разработки мобильного приложения для IOS используются языки Objective-C и Swift, в то время как для Android приложений используется Java. Из-за необходимости создания мультиплатформенного приложения, возникают трудности. В особенности с платформой IOS, так как для комфортной разработки приложения требуется ПК работающий на операционной системой macOS.

Поэтому необходимо найти подход, который бы позволил комфортно разрабатывать приложение для IOS на операционной системе windows. В ходе

поиска инструментов разработки была найдена JavaScript платформа React Native, которая использует компонентный подход для обеих основных платформ одновременно. Компоненты разработаны таким образом, чтобы один и тот же компонент работал на IOS и Android, т.е. появляется возможность писать один код и использовать один язык для обеих платформ. Так же платформа примечательна тем, что она, используя синтаксис React, позволяет перейти с разработки веб-приложений к разработке мобильных приложений и наоборот.

React – это JavaScript платформа, которая позволяет разрабатывать быстрые и отзывчивые интерфейсы веб-приложений. React был создан компанией Facebook и первая версия была доступна в марте 2013 года [3]. Изначально React планировалось использовать только для создания веб-приложений, но позже был представлен React Native.

В основе React лежит несколько концепций: компонентный подход, односторонний поток данных, виртуальная объектная модель документа, синтаксис JSX.

UI (User Interface) состоит из компонентов, которые изменяются в зависимости от состояния и передаваемых им свойств.

Свойства передаются к компоненту от его родительского компонента. Компоненты получают свойства как набор неизменяемых JavaScript объектов. Когда изменяется значение свойства, функция отрисовки компонента обрабатывает и показывает внесённые в интерфейс изменения.

Подход виртуальной объектной модели состоит в использовании легковесной копии обычной объектной модели документа [4]. Если необходимо изменить элементы интерфейса, то все изменения сначала производятся с виртуальной моделью документа. Новое состояние виртуальной модели сравнивается с предыдущим состоянием и выявляется разница. На основе этой разницы React оптимально вносит изменения в соответствующие элементы обычной объектной модели. Это решает проблему производительности связанную с изменением большого количества элементов объектной модели].

Если мы напрямую взаимодействуем с объектной моделью, то мы последовательно изменяем каждый элемент, в то время как React сравнивает что изменилось через легковесную виртуальную объектную модель и только после этого вносит единовременное изменение объектной модели документа. Этот подход не только помогает в плане производительности. Существующую виртуальную модель можно адаптировать под разные платформы, как в итоге это и было сделано в React Native.

Таким образом, React Native хорошо подходит для разработки мобильных приложений, так как упрощает и удешевляет процесс разработки в ходе использования уникального подхода для мультиплатформенной разработки.

Список использованной литературы:

1. Working With Expo [Электронный ресурс] // Expo. 2018. URL: <https://docs.expo.io/versions/v27.0.0/workflow/> (дата обращения: 10.марта.2020).
2. React. A JavaScript library for building user interfaces // reactjs. 2018. URL: <https://reactjs.org/> (дата обращения: 9.марта.2020).
3. Optimizing React: Virtual DOM explained. // evilmartians. 2018. URL: <https://evilmartians.com/chronicles/optimizing-react-virtual-dom-explained> (дата обращения: 8.марта.2020).
4. Introducing JSX [Электронный ресурс] // reactjs. 2018. URL: <https://reactjs.org/docs/introducing-jsx.html> (дата обращения: 11.марта.2020).
5. Build native mobile apps using JavaScript and React [Электронный ресурс] // Facebook. 2018. URL: <https://facebook.github.io/react-native/> (дата обращения: 11.марта.2020).

© Е.В. Зотов, Е.С. Занина, 2020