

COMPARATIVE ANALYSIS OF TWO MOST POPULAR JAVASCRIPT FRAMEWORKS FOR MOBILE APPLICATION DEVELOPMENT

Ten T.L., Chernov S.S.

Karaganda Economic University, Karaganda, e-mail: chernov.home@gmail.com

This article presents a comparative analysis of two of the most popular Java Script frameworks for development of a mobile application such as Ionic 2 and React Native. At the very beginning, necessity of having mobile applications in modern times was considered. After that, the article explained a way mobile applications are distributed among their consumers. Then the author continues with a general explanation of the main functions of each of the frameworks. Furthermore, each framework's features will be discussed. Moreover, the way of writing the code using each of the tools will be also explained. Then we will focus on the debugging process for both Ionic 2 and React Native; look at the way how to work with styles; we will consider how the precise tool influences the size of the final application. Next, frameworks will be compared depending on their popularity. Finally, the conclusion will be provided in the form of a list of the main advantages and disadvantages of compared tools.

Keywords: mobile application, front-end framework, information systems, Ionic 2, React Native

The need for mobile data access speeds across the company's business processes. Mobile coordination of reports or approval of vacations is already a must have component of the work process in corporations. Every employee has a mobile device, so the company can use these devices without investing in them. A large variety of mobile devices must be considered when developing a corporate mobile application: it should work and have the same appearance on all devices. In this article, I will consider the creation of such corporate mobile applications using various Java Script frameworks.

Corporate mobile apps are usually distributed through the company's internal app stores. This makes it possible to control access to applications within the company, as well as forcefully install application updates. For the development and delivery of corporate mobile applications, an appropriate infrastructure, known as the Mobile Enterprise Application Platform (MEAP), is required.

For MEAP, there was conducted an analysis and a comparison of existing platforms and vendors by such leading companies as Gartner and Forrester (Gartner Magic Quadrant and Forrester Wave, respectively). All the leaders of these comparisons are considering the possibility of creating native and hybrid mobile applications. Native applications are written using specific platform programming languages (Swift or Objective-C for iOS, Java for Android). Hybrid mobile applications are written in JavaScript using various libraries and frameworks.

Using JavaScript frameworks allows us to create new mobile applications using the existing front-end development team. For the company, this means that the same team

can perform different tasks – website development, mobile application development. Therefore, it appears to be pretty beneficial for the both sides.

Among the leading frameworks, two of the most popular frameworks can be identified: Ionic 2 and React Native. Further, each of them is considered separately, and their comparison is carried out. The comparison was performed only for two operating systems: iOS and Android.

Ionic Framework is designed for creating hybrid mobile applications. It includes a set of Java Script and CSS components created on top of Angular 2, SASS and Apache Cordova. The history of this SDK begins in 2013, when the company Drifty Co. decided to create its own infrastructure for the purpose of writing hybrid applications, which was supposed to be focused on performance and built using modern web standards. Release Ionic 1 was held in May 2015. Version 2 was released in 2016. The main difference between the second release and the first one is the transition from Angular 1.x to Angular 2.x.

At its core, the Ionic Framework is an add-on over the very popular Apache Cordova framework, but with its powerful CLI (Command Line Interface) and extensive documentation. Following the principles of Apache Cordova, applications built with the Ionic Framework are hybrid HTML applications. Such applications are run in a special shell on devices (UIWebView for iOS and WebView for Android), which allows you to display HTML and execute JavaScript. Accordingly, when working in the application, the user works as if in a web browser [1].

Apache Cordova is a kind of layer between the user interface and device resources. Any

user interface widgets or MV* (Model-View) frameworks are not included. If you need to use UI widgets and / or MV* framework, then you need to select and include them into the application yourself, as third-party resources. Ionic 2 is one of the frameworks that provides UI widgets. The Angular 2+ framework provides the MVC model for it.

Native phone features (for example, camera, key store, GPS coordinates) are not accessible from a web browser. Therefore, Apache Cordova plugins are used to work with them. In addition to the official plug-ins, there are a number of third-party open source plug-ins.

React Native

React Native is a JavaScript framework developed by Facebook for creating native mobile applications using JavaScript. The history of React Native began in 2013 with Facebook's internal hackathon project. React Native was first publicly mentioned at React.js Conf in January 2015, in March 2015 at F8 it was announced that React Native is available on GitHub. The framework is built on top of the ReactJS library. React Native uses JavaScript API on top of native components. During the implementation of an application, developer writes JavaScript code that works with the native components of the operating system. That is, React Native uses the same fundamental standard UI blocks as regular iOS and Android applications without using either a browser or WebView / UIWebView [2].

An application on React Native uses a separate thread to execute javascript on the device. This thread interacts directly with native code. It is important to mention that native components (including user interface elements) are different for various operating systems. And, accordingly, for each operating system it is necessary to create its own "bridges" between the JavaScript code and these components. A number of such components are provided by Facebook, and there are also many freely available components developed by the open source community.

Ionic 2 vs React Native

Both Ionic 2 and React Native allow you to create mobile applications. The fundamental difference between these frameworks lies in the type of mobile application that is created using one or another framework. Ionic 2 allows you to create a hybrid mobile application that renders in WebView / UIWebView. If you need to use the platform API, you will need to use additional plugins. React Native on the other

hand allows you to create a native mobile application using the fundamental UI building blocks (like regular iOS and Android applications that are written in the native language Objective-C / Swift and Java, respectively). The application on React Native uses certain templates defined by the platform itself.

Application code

For creating an application on Ionic 2, the Angular 2 framework and Apache Cordova are used. The application can be written in JavaScript or any other language that is broadcast in JavaScript. The most popular typed JS superset is TypeScript, introduced by Microsoft in 2012 and positioned as a web application development tool that extends JavaScript capabilities, and most of the examples in the Angular 2 documentation are implemented in it. [5] Angular 2 implements the dependency injection design pattern (software design pattern), which consists of 3 elements:

- Injector – provides API access for dependency injection;
- Provider – describes how to create a dependency instance, has a marker pointing to the factory function that creates the object;
- Dependency – the type to which the created object belongs.

In addition, Angular 2 uses a component approach that allows you to separate business logic and things related to the UI. In addition, the separation is also used at the level of the project elements: HTML, JavaScript and CSS are kept in separate files.

In order to start writing an application on React Native you need to get acquainted with ReactJS, JSX. It is also recommended to be familiar with Redux and EcmaScript 2015 to speed up development, but knowledge of these components is optional.

You can also use Flow – this is a static code analyzer and a set of syntactic structures to directly indicate the type of a variable. Flow can calculate the type of a variable, without making changes to the code (unlike TypeScript), which allows you to start using it now in any project. It is possible to specify types in the TypeScript style independently.

In the process of writing an application using React Native, you create components that are self-sufficient, each component is responsible for changes in its state. There are stateless and stateful components. Components that do not have a state can pass properties down the component structure (props are set by the parent component and are fixed during the entire life cycle of the component) [3].

Debugging process

Debugging applications in Ionic 2 is standard for front-end developers. After running the application on the emulator / device, it is possible to debug applications using Safari web inspector (iOS only) or Chrome Developer tools. These tools provide access to HTML, CSS, and JavaScript. During debug, you can quickly edit styles, search for nodes in html, execute and watch the result of JavaScript functions, as well as network requests. In addition to running on a device or an emulator, in the case of Ionic 2, it is also possible to launch the application directly in the browser (ionic serve). This can speed up development, and with the help of the ionic serve --lab command, it is easier to test applications on different screen diagonals and platforms.

Ionic 2 has the Hot Reload option. It allows you to make changes to the code and see these changes on the device / emulator without rebuilding. However, it is not always possible to use it. Due to the fact that application is working in a web browser when using http requests, we encounter a CORS problem.

To debug applications on React Native, it is also possible to use Chrome Developer Tools. If development is conducted on Apple devices, then it is possible to use the Safari web inspector for an emulator or an iOS device. To view the items, you can select the "Show inspector" option in the Emulator / Device Developer Menu. At the same time, you can see the styles of elements, but changing them is possible only in the IDE (the method for quickly editing styles in developer tools does not work here). But using LiveReload or HotReload options within a few seconds, the saved changes will be reflected in the emulator / device. In case of global changes made, it is better to rebuild the project. Also, there is the Network Tab in the Developer Menu, which allows you to track HTTP requests. Here you can see the type of request, method, url, and more. It is kind of unusual if you got used to debug in the browser. For debugging,

you can also use IDE for each of the platforms – Android Studio and Xcode, that allows you to inspect in detail all the logs and operation of the application (resource consumption, speed, etc.) [4].

Styles

Ionic 2 is written HTML5 code and uses the familiar front-end developer SASS. There are no restrictions on dimensional units, the SASS preprocessor is used, you can also use flex markup. I often had to redefine Ionic 2 variables. But sometimes this was not enough, it was necessary to study the final html in the browser, see what additional elements were generated and what styles they had in order to understand what and where to redefine or add. Also, a significant difference was in the display of elements on different platforms. What looked good on iOS devices sometimes broke styles on Android devices, it was necessary to find compromises.

In React Native, the creation of styles differs from the usual for frontend developer. Here css-detailed syntax is used, which is written as a js object. Then this object with "styles" is transformed into understandable for the platform instructions on the location and coloring of elements.

Application size

For both frameworks, there are "empty" applications – these are application frameworks with minimal code. Such applications are generated using the CLI:

- Ionic 2: ionic start myApp blank --v2
- React Native: react-native init AwesomeProject

Empty applications were collected in two modes: debug and release. The first mode is designed for debugging the application, the second – for the direct delivery of ready-made applications to end users.

Applications on different frameworks have different sizes on different platforms. Ionic 2 wins on Android, and React Native wins on iOS.

Application builds sizes in different modes

	Ionic 2		React Native	
	Application in debug mode	Application in release mode	Application in debug mode	Application in release mode
IOS	18.3 Mb	17.9 Mb	4.8 Mb	4.6 Mb
Android	5.54 Mb	4.0 Mb	32.38 Mb	16.35 Mb

Popularity and activity of the community of the frameworks

When comparing the popularity of frameworks, you need to consider that React Native is a younger technology. Initially, React Native had support for iOS only. Android support in React Native appeared only in August 2015, and the Ionic Framework at that time was developing for more than two years (although it had version 1.0) and had the support of all popular platforms.

Popularity and activity were compared by the two most popular resources: stackoverflow and github.

For the Ionic Framework on stackoverflow, there are about 60,000 results for Ionic and Ionic 2. When considering statistics on the Github repository, it is important to note that in mid-2016, the Ionic 2 code migrated to the Ionic repository. In the Ionic repository, more than 200 participants made about 6000 commits in 15 branches with 88 releases. About 700 open questions and ~ 9000 closed questions are in the repository of the project.

For React Native on stackoverflow, there are about 25,000 results for react-native. Statistics on the Github React Native repository show that this framework is more active than Ionic 2: over 1200 participants made about 10,000 commits in 60 branches with 160 releases. The project repository contains ~ 1100 open questions and ~ 7000 closed.

Conclusions

Corporate applications for mobile devices, created using JavaScript, have occupied their niche. The speed of such applications is sufficient compared to applications written in native platform languages. In general, one cannot say that some kind of framework is better, and some kind is worse. They are just different. The choice depends on the tasks, the ultimate goal and many other factors. In general, we can recommend using Ionic 2 for rapid development of prototypes of mobile applications, and React Native – for the full development of ready-made solutions. Below is a list of the advantages and disadvantages of each framework, based on our impressions of using Ionic 2 and React Native.

Advantages of Ionic 2:

- a well-known set of tools – for a frontend developer, the set of technologies used is familiar, it significantly reduces the time to complete the task

- quick start – using the application templates provided to us by Ionic, you can create a prototype in a short time to show to the customer (we are talking only about the methods of the

frameworks described on the official websites, third-party Starter Kit were not considered by us)

- “Write once, run anywhere” – the written application works on all popular platforms, it gives cross-platform with minor code changes

Disadvantages of Ionic 2:

- stylization – changing the default styles to bring them in accordance with the layout significantly increases the time spent on creating and testing applications on different platforms and devices

- Browser use – longer application load and poor responsiveness when the CPU is loaded due to using the browser shell to simulate the behavior of the components of each platform

- native functions – when a native event occurs, the main thread is blocked and control is passed to the JavaScript code, waiting for its instructions, which can lead to unpredictable operation of native functions

Benefits of React Native:

- standard native UI blocks – the use of the fundamental standard UI blocks, like regular iOS and Android applications, familiar to the user from other native applications, facilitates user interaction with the application

- non-blocking javascript execution – javascript runs in a separate background thread, interacting with the main thread asynchronously

- styling – the application looks equally good on different platforms and devices

- load on the CPU – the tests carried out by the application using the CPU and memory show better results compared to the Ionic 2.

Disadvantages of React Native:

- the need to work with native code platform – it may be necessary to understand the components written in ObjectiveC / Swift or Java

- complexity of the start (without experience with ReactJS) – additional time is needed for getting to know React, JSX and a different writing approach (components as separate interface units that are responsible for their own state and behavior).

References

1. Official Ionic web site [digital source] URL: <https://ionicframework.com/docs/v1/guide/preface.html> (date of treatment 07.08.2019).

2. React native handbook [digital source] URL: <https://medium.freecodecamp.org/the-react-handbook-b71c27b0a795> (date of treatment 07.08.2019).

3. When to Choose React Native for Building a Mobile App and When to Avoid It [digital source] URL: <https://rubygarage.org/blog/react-native-for-mobile-app-development> (date of treatment 07.08.2019).

4. React Native – Is it Really the Future of Mobile App Development? [digital source] URL: <https://hackernoon.com/react-native-is-it-really-the-future-of-mobile-app-development-31cb2c531747> (date of treatment 07.08.2019).

5. App Development: Ionic or NativeScript? [digital source] URL: <https://codeburst.io/app-development-ionic-or-nativescript-40b118bd8e51> (date of treatment 07.08.2019).