

---

# COMPARISON BETWEEN DIFFERENT JAVA-SCRIPT FRAMEWORKS FOR CROSS-PLATFORM MOBILE APPLICATION DEVELOPMENT

---

Edited by:

Name: Agbele Kolade, Matric NO: 26623

Name: Adeyeye Ayodeji Daniel, Matric NO: 28003

*University of Applied Sciences Rhine Waal*  
*Hochschule Rhein Waal*

# Abstract

Mobile App development continues to rule the world. With smartphones having become nothing closer to than a personal asset to every individual, it is pretty obvious that Mobile App development is here to stay. Not just stay but stay long and strong. Smartphones have been built to operate on a couple of different platforms, the prominent ones among them are the Android, Windows and iOS platforms. Subsequently, the mobile apps developed to work on these platforms, also use different languages. For e.g. Android compliant apps are developed using Java, Windows compliant apps are developed using .Net and iOS platforms use Object C. However, since its introduction, nearly two decades ago, JavaScript has emerged as the most popular and sought-after programming language for mobile apps development today. Statistics indicate that JavaScript scores heavily and is currently being used by more than 94% of all the websites. Where JavaScript scores, is that it helps the mobile app developer to develop dynamic and interactive web pages, which are client central and deploy custom client-side scripts. The aim of this research is to make a COMPARISON BETWEEN DIFFERENT JAVA-SCRIPT FRAMEWORKS FOR CROSS-PLATFORM MOBILE APPLICATION DEVELOPMENT (Ionic, Native scripts and React native) and we will be evaluating each of this frameworks with respect to Performance, Costing, Ui and theming, Game development, Native component, Popularity by GitHub stars, Developed by and Cross platform support. The major similarities between these frameworks is that they all support different platforms which are IOS and Android. We noticed the performance level of each framework varies depending of the size and purpose of which the application is to be developed.

# Introduction

There are more than two billion smartphone and tablet users worldwide. According to latest trends, it is being expected that anything and everything will be available as a mobile application. This brings another major question into mind of an application developer or an organization that which technology/platform should they use to develop their mobile application to attract users or provide them a better functionality along with availing maximum features provided by the mobile platform.

Developing effective solutions for mobile applications involves multiple disciplines and a wide range of technologies. Mobile applications should be engaging, deliver a consistently meaningful user experience across different devices and work seamlessly both online and offline. This may sound simple in theory, but is in fact highly complex, technically, due to a range of factors such as the highly fragmented mobile technology landscape, rapidly evolving standards, limitations imposed by the mobile device itself (screen size, input methods, display capabilities, etc.) and also constraints of the mobile network such as high latency and low bandwidth.

As mobile applications become increasingly popular and the technology moves into mainstream, the demand for sustainable practices and tools for building and supporting ongoing development becomes apparent. Although there has been lot of effort within the community to promote best practices and mobile standards which has led to the release of consistent guidelines and frameworks to address common problems like the delivery of cross-platform content that works seamlessly on any device. Since the release of NodeJS the use of java scripts for development of mobile applications has increased with top organizations and individuals releasing different mobile application frameworks to solve the problem of cross platform development. This Report aims at comparing these technologies and how effected they have been and how widely they have been used in the development of mobile applications.

## Related Work

Much related work can usually be identified for an article that compares various technologies. However, if it deals with cutting-edge technology, the number of similar papers shrinks dramatically. General papers on the technologies dealt within this paper are cited in the appropriate sections, particularly in Section 3. Thus, this section assesses existing work that compares two or more approaches for cross-platform app development. Until recently, papers only discussed mobile platforms – or rather operating systems – for mobile devices. An example is the paper by Cho and Jeon (2007). Comparison papers such as by Lin and Ye (2009) only marginally help developing multiplatform apps. The same applies to very specialized papers. They usually rather concern the business perspective than deal with technology. An example is a study of mobile service platforms (Tuunainen et al., 2011). But even technically-driven papers that address multiple platforms do not necessarily help to develop cross- platform apps. For instance, a study of smartphone malware (Felt et al., 2011) only roughly hints to platform particularities. Anvaari and Jansen (2010) have compared the predominant mobile platforms with regard to the openness of their architectures. Their approach takes a very close look at one aspect and thus can be seen as complementary with our work. Charland and Leroux (2011) compare the development of native apps and Web apps. In contrast to our approach, they do not take a cross-platform perspective. A comparison of iPhone and Android development is presented by Goadrich and Rogers (2011). Despite the topic, which is similar to our work, their aim is different. In fact, they try to answer which platform should be used for the education of students. Another study deals with mobile cloud apps (Lakshman and Thuijs, 2011). While the authors deal with cross- platform development, they focus on native thin clients that access cloud services. A number of publications address more than one platform (David, 2011; Anderson and Gestwicki, 2011; Firtman, 2010). While these publications foster a better understanding of the platforms, they do not really compare the different approaches. Rather, they explain how to use a technology on a multitude of platforms or devices. Due to the high relevance for practitioners, the topic is also recognized in technology weblogs (Newman, 2011; Behrens, 2011). Although such articles give valuable advice, they cannot be compared to our structured approach.

## 2.0 Cross Platform Frameworks

### 2.1 Definitions

Platforms are a combination of hardware (system plus any possibly built-in sensors or actuators), operating system, vendor-provided software SDKs and standard libraries, which together offer the base for building software for that platform. A framework allows the reuse of a usually predefined application architecture and provides components that help in quickly setting up an application. Thus, a cross-platform framework allows to reuse parts of the application source code for multiple platforms and may additionally offer building blocks such as an architecture style (e.g. MVC2), user interface API or other functionality that is not specific to a single platform. Such frameworks can also expose APIs for platform-specific features. Mobile applications exploit features specific to mobile devices and platforms, for instance a camera, accelerometer or fine-grained location retrieval. Pure web applications running in a mobile browser do not yet have access to all device features through a JavaScript interface and hence do not fall into the category of mobile applications.

### 2.2 Types of cross-platform java scripts frameworks.

There are numerous Java scripts mobile frameworks but we will only be focusing on the three frameworks which are Ionic, NativeScript and ReactNative.

#### **Ionic**

Ionic is a commonly used open-source framework to build hybrid mobile applications. This framework provides useful tools and services which developers can utilize for creating native and progressive web application seamlessly. With Ionic, developers need fewer efforts and also can minimize maintenance expenses.

1. Code works across platforms and devices.
2. Single programming language for all operating systems.
3. Variety of Cordova plugins available.
4. Utilization of well-known web technologies.
5. Application testing is easier.

#### **NativeScript**

NativeScript is a creative hybrid application development framework, it is extremely useful in building native cross-platform applications with super performance. In addition, NativeScript allows developers to reutilize skills from Angular, JavaScript, and TypeScript. Its holistic approach lets developers write once and run it everywhere.

1. Single code base and then develop it for Android or iOS.
2. Gives faster access to native libraries.
3. Team Native Script welcomes feature requests to enhance.
4. Developers can learn Native Script using JavaScript, XMS, and CSS

## React Native

React Native is an open-source mobile application framework created by Facebook. It is used to develop applications for Android and iOS by enabling developers to use React along with native platform capabilities.

1. Developers can seamlessly copy half the code to build apps on other platforms.
2. With native components, users can get native app experience.
3. Revamping complete code is not required.
4. Developers can easily run the code everywhere by putting fewer efforts.

## 3.0 Materials and methods

### 3.1 Methodology

The decision for the three selected cross-platform frameworks was mostly based on active development, stability and popularity of the frameworks. In order to gain an insight on the frameworks, we read through the official documentations for information and reports from other developers on stackshare.io. These evaluations were based on the following criteria's Performance, Free and open source, Ui and theming, Game development, Native component, Popularity BY GitHub stars, Organization responsible for the Development and maintenance and Cross platform support.

## 4.0 Results

Table 1 below summarizes the results of the evaluation. The table shows the main features of each of the frameworks. However, the actual differences in each category need to be considered.

	Ionic	Native scripts	React native
Performance	Not suitable for high performance applications	It is suitable for high performance applications	It suitable for high performance and enterprise applications
Free	Yes	Free and open source	Free and open source
Ui and theming	Not suitable for UI intensive applications	Angular JS and Vue	React UI markup
Game development	Not suitable for game development	It is suitable for Game development	It is suitable for Game development
Native component	It uses HTML, CSS AND JavaScript	It uses Native UI markup CSS AND JavaScript	It uses React UI markup, native component, CSS AND JavaScript
Popularity By GitHub stars	40,000+	18,000+	84,000+
Developed with and maintained by	Angular JS and Apache Cordova and maintained by individuals	Angular JS and Vue.js and maintained by Telerik and Google AngularJS team	ReactJS and maintained by Facebook
Cross platform support	IOS and Android	IOS and Android IOS	IOS and Android

Table 1: Overview of the evaluation for the frameworks

## **4.0 Discussion**

### **4.1 Ionic**

Ionic is an open source framework which is not suitable for high performance applications, Game development and UI intensive applications. However, it is easy to learn and implement because it has no native UI markup and it uses HTML, CSS and JavaScript. Moreover, Ionic framework compare to the other two frameworks we are considering in this report, Ranks second with over 40,000 stars on Github. Ionic runs on Angular JS and Apache Cordova, at the same time it supports IOS and Android.

### **4.2 Native scripts**

NativeScripts is suitable for high performance applications which supports IOS and Android platforms, it is free and open source. It also uses Angular JS and VueJS UI markup which makes it suitable for Game development. Furthermore, NativeScripts ranks the least among the three frameworks with a Github star of over 18,000+. It uses either Angular JS and Vue.js, which are maintained by Google AngularJS team and Telerik respectively.

### **4.3 React native**

React native is an open source framework which is suitable for high performance, enterprise, Game development, UI intensive applications and supports IOS and Android. However, it is easy to learn and implement because it uses React UI markup, native component, CSS AND JavaScript. In addition, React native ranks the highest with over 84,000 stars on Github and it is maintained by Facebook.

### **4.4 Conslusion**

Conclusion The major similarities between these frameworks is that they all support different platforms which are IOS and Andriods. We noticed the performance level of each framework varies depending of the size and purpose of which the application is to be developed.



# Bibliography

1. <https://www.developeronrent.com/blogs/javascript-frameworks-for-mobile-app-development>
2. <https://medium.com/@agrawalsuneet/mobile-development-native-or-cross-platform-f63872ae9f88>
3. [http://cdactvm.academia.edu/Departments/Wireless\\_and\\_Mobile\\_Computing/Documents](http://cdactvm.academia.edu/Departments/Wireless_and_Mobile_Computing/Documents)
4. <https://docplayer.net/8424928-Mobile-app-infrastructure-for-cross-platform-deployment-n11-38.html>
5. <https://www.scribd.com/document/141833009/Cross-Platform-Mobile-Development-sxasa-Cross-Platform-Mobile-Development>
6. <https://www.scribd.com/document/141833009/Cross-Platform-Mobile-Development-sxasa-Cross-Platform-Mobile-Development>
7. <https://docplayer.net/6943656-Cross-platform-mobile-development.html>
8. [https://ase.in.tum.de/lehrstuhl\\_1/research/paper/sommer2013crossplatform.pdf](https://ase.in.tum.de/lehrstuhl_1/research/paper/sommer2013crossplatform.pdf)
9. [https://mafiadoc.com/evaluation-of-cross-platform-frameworks-for-mobile-applications-tum\\_59753c831723dd0e4ec7c930.html](https://mafiadoc.com/evaluation-of-cross-platform-frameworks-for-mobile-applications-tum_59753c831723dd0e4ec7c930.html)
10. <https://ytavivuvavuj.tk/sencha-touch-2-mobile-javascript-framework.php>
11. <https://www.redbytes.in/javascript-frameworks-for-mobile-app-development/>
12. [https://en.wikipedia.org/wiki/React\\_Native](https://en.wikipedia.org/wiki/React_Native)
13. [https://www.researchgate.net/publication/259852798\\_Evaluation\\_of\\_cross-platform\\_frameworks\\_for\\_mobile\\_applications](https://www.researchgate.net/publication/259852798_Evaluation_of_cross-platform_frameworks_for_mobile_applications)