



[Mobile application development](#) is kind of a big deal, ever growing and expanding, and if you're not using the right tools to help optimise your idea, you might well be setting yourself up for disaster. Nowadays we have many tools at our disposal for developing apps and they are much easier to work with than dealing directly with the code. Learn more about how front-end frameworks operate and 25 of the top tools to consider in your app development process.

Understanding Front-End JavaScript Frameworks

Back in the day, developing a mobile app was a bit of a challenge, to say the least: In order to get to the point where your app would enter the development stage, you'd first need an idea of what your app would be like on screen. After you had the basic idea down, a front-end designer would encrypt your image with HTML, after which it would be converted (by a developer) into a working web development page, but there's a catch!

Your idea and the final result after the developer has worked his magic would rarely look the same, even more so when you had a single-page web app!

This is where JavaScript frontend frameworks come in to save the day. These frameworks formulate web apps and create an environment where design and backend workflow can be easily coded.

Finally, designers can play around with ideas in a way that won't be affected by the whole development process. In short: We now have a solution that simplifies the process and costs you a lot less valuable time!

There are numerous frameworks that you can use for mobile app development, but today, we'll show you 25 of the best. Let's dig in...

25 Best Mobile App Development Frameworks

1. [AngularJS](#)

AngularJS is arguably one of the most popular JavaScript frameworks, and as far as Model View Control based apps go, it has everything you'll ever need. AngularJS works on the concept of two-way data binding (and happens to be the pioneer of model view controller architecture for web apps).

2. [EmberJS](#)

Just because it's slightly less popular than AngularJS doesn't mean it's any less powerful. With a robust routing framework, EmberJS is a great option if you've got complex nested routes and templates within your page.

3. [CanJS](#)

Although it's the new kid on the block, CanJS is impressive when it comes to ease of use and its flexibility. The community is a lot smaller than you'd find on the big name frameworks like AngularJS, and so are the learning materials that apply to the JS framework. Another thing you'd like to keep in mind is the fact that CanJS needs Zepto libraries to run efficiently.

4. [KnockoutJS](#)

Popular among JS web app designers, KnockoutJS is a lightweight toolkit that makes for a perfect fit if you want to develop your own framework. KnockoutJS is easy to get up and running, and also easy to work on, but without built-in features to convert plain objects into nested observables, you'll have limited abilities to properly structure your app.

5. [BackBoneJS](#)

Another great partial framework that a lot of developers rely on when they're building their own frameworks is BackboneJS. BackboneJS is awesome to use, but just keep in mind that this toolkit was made with advanced developers in mind, so it requires quite a big learning curve to master.

6. [jQuery Mobile](#)

This is a free framework you can use in order to develop cross platform apps, and quickly too! JQuery's great UI across platforms like iOS, Android, Symbian, BlackBerry, WebOS, and Windows Phone (to name but a few) is what the framework is best known for.

7. [Sencha Touch](#)

Arguably one of the best frameworks amongst developers, Sencha Touch was the pioneer HTML5 framework which could be used for smartphones and tablets alike. A full set of themes for mobile platforms, enriched UI, the awesome MCV software architecture pattern, and an extensive data package are just some of the great developer-friendly features that you'll love about Sencha Touch.

8. [PhoneGap](#)

Most commonly used to develop native hybrid apps for mobile operating systems, PhoneGap is a framework built in emulation that has an open source WYSIWYG IDE. It is based on Cordova and uses JavaScript API's to connect with a device's native functioning.

9. [React.js](#)

This is the framework you'll turn to help you create the UI of your projects, which explains the reason why it's such a popular framework. Front-end component development is simplified with the use of React.js, and as such, platforms like Facebook and Instagram are using this framework to their advantage.

10. [Spine](#)

Based on the MVC model-view-controller application design architecture, Spine is a framework that was inspired by BackboneJS. This framework can help your backup lightweight controller implementations, but keep in mind that its related library has been written in CoffeeScript, and depends on jQuery to run effectively.

11. [Zepto.js](#)

As an open-source framework compatible with jQuery, Zepto.js is best used when you have an app to develop that's being built specifically for mobile WebKit browsers like Safari and Chrome. Zepto.js is best known for its incredible, lightweight codebase, which makes it easy to use, even more so, if you know jQuery since the architecture is similar.

12. [Enyo](#)

Enyo is an object-oriented application framework that is free and requires HTML5 and JavaScript to develop mobile apps, which means it enables apps to become easily accessible across platforms like Android, BlackBerry, and iOS, to name a few. With Enyo, you'll get features like internationalisation and localisation.

13. [Ratchet](#)

Initially used by Twitter to create mobile app prototypes, Ratchet became globally recognised after being introduced as an open source project for app developers across the world. The latest (v2.0.2) Ratchet release offers you an abundance of JavaScript plugins and UIs. It also supports Chrome and Android's stock browser, plus Safari in iOS v7.1+.

14. [The M Project](#)

The M Project's v2.0 Absinthe is a rock-star JavaScript HTML5 framework. Based on BackboneJS, The M Project offers you a number of different themes, URL navigating and deep linking, 60+ CSS transitions and Sass support. It's a stable framework across Android, iOS and Kindle Fire platforms,

plus it contains core files to build jQuery mobile apps.

15. [Kendo UI](#)

Kendo UI is a cross-platform framework that packs everything you need for building mobile and web apps when using HTML5 and JavaScript. It features over 70 jQuery-based UI widgets, support for Bootstrap, offline data solutions, mobile controls, integration with AngularJS, and a bunch of other developer-friendly features, which is why it's ranked as a top mobile app development framework.

16. [Jo](#)

If you're working on HTML5 enabled browsers like iOS and Android, Jo is a framework you'll love. With its native UI elements like a web form widget for login screens and a popup widget for adding more info when users click on an interface element, Jo is easy to use and very effective.

17. [Wijmo](#)

Wijmo is a framework that packs over 40 HTML UI Query widgets for mobile and web app development, which makes it perfect for interactive designs. This framework is built of jQuery UI and jQuery Mobile, which gives you the advantage over common standard usage of HTML5.

18. [SproutCore](#)

SproutCore is an HTML5 Application Framework, but it's more than just a plugin; it provides you with a native experience. With a smooth-scrolling list of over 50,000 items, an organised and maintainable application code, and great business logic, SproutCore is definitely a top web app development framework.

19. [NimbleKit](#)

As far as preferred frameworks go for developing iOS applications, NimbleKit is arguably one of the top options to turn to. The great thing about this framework is the fact that you don't require an extensive knowledge of Objective-C or the iOS SDK to work with it, you just need to know how to write an HTML page with JavaScript code!

20. [Corona](#)

Corona is a software development kit framework that utilises Lua and C++ along with OpenGL to build apps for NOOK, Android, iOS and Kindle Fire while allowing for mobile application development cross-platform with just the one code.

21. [Xamarin](#)

Xamarin is an advanced framework recommended for development companies who operate remotely. This C# based platform enables the calling of native APIs in the resulting mobile applications. The code execution is on a .NET framework at the runtime, and it gives a native dialect of either Android or iOS.

22. RhoMobile

RhoMobile is an ROR-based integrated framework which enables developers to create apps for Android, iOS, Symbian, Windows Phone, and RIM. This framework packs many different developer-friendly extra features, which makes it easy to use and very versatile.

23. Adobe Flex

As an open source application framework, Adobe Flex is a toolkit that's best used for creating and storing expressive web apps that install on major browsers, desktops, and devices. Adobe flex uses a programming language that has a model which is suitable for developers across different backgrounds, so you don't need to be experienced to use it.

24. Unity 3

This tool is a development platform that's best used for creating multi-platform 2D and 3D games, but it's also great for interactive experiences. Unity's comprehensive ecosystem is easy to use, enables the creation of high-end content, and provides the opportunity of connecting with your audience.

25. Mono Sync

Supporting C, C++, PHP Java, JavaScript, Python and ROR, Mono Sync is a powerful and versatile framework. With Mono Sync, the native mobile applications can be transformed for other platforms like Symbian, Linux Mobile, iOS, Android and Windows Phone, without any issues.

How to Choose the Best Framework for Your Company

You might have been looking for a one-shot solution for your mobile app development, but since all businesses vary in their requirements, what works for one company might be a total flop for another. You must weigh up the pros and cons that each mobile app development framework brings to the table, keeping your app's actual purpose in mind, and then make a decision based on your company's specific needs that will make your decision a no-brainer.