

Native Web Apps

The web platform has evolved. Browsers are no longer just document renderers, they have become the most capable and ubiquitous application runtimes on the planet.

Right now, the most common term for applications that use the browser as a runtime is "Single Page App". I've even heard people pronounce it like the word "spa". If you've done this, we can probably still be friends :)

Personally, I've always disliked the term "single page app" because it fails to describe what my applications are. The fact that the app consists of a single initial HTML page is not what makes it interesting or unique. In fact, often times it's plain wrong. A well-built "single page app" updates the URL as the user navigates around to different "pages" within the app, keeping their place if they refresh or share that URL. Perhaps at one time the "single pageness" was its defining characteristic, but these days the term makes it sound more like you're building an application that doesn't respect the URL and thus breaks the web.

When I was first writing [Human JavaScript](#) I struggled to find a good word for these apps that I could use throughout the book. Each of them has issues, here are a few in particular:

Browser app

I nearly picked this one. It's good, but it fails to cover hybrid and other apps that run in a web-like runtime that isn't a browser. As I've written previously, I believe [the web has outgrown the browser](#) and also, it can possibly be confused with creating something like [a chrome app](#).

Clientside app / clientside web app

This is pretty good too, but also falls short. The name implies that there's

also a server-side component and doesn't make any distinction about what type of client it is. Inserting "web" in the middle helps that, but it still somehow fails to capture what it is.

JavaScript App

Node.js exists, need I say more? There are plenty of javascript apps that do everything else under the sun, like [control robots](#) for example. This simply isn't specific enough.

HTML5 app

Not bad, but again, to many of us anything that has "HTML" in it, is strongly associated with static markup, when clearly the thing that makes the apps we're discussing unique is their dependence on the web runtime.

What is their defining feature?

The thing these apps *all* have in common is that they all depend on the native web technologies: HTML, CSS, and JavaScript (arguably, you could add WebGL to that list).

In the modern web these are not enhancements, they *are* the native technologies.

Therefore I think it's time we start calling them what they actually are: **Native Web Apps**.

Why "Native Web App"?

- The exact capabilities of each of the individual web technologies is not what matters. This term works whether your markup is HTML 4.01, HTML5 or HTML37. It doesn't matter, and the term won't be obsoleted by the next version of any spec.
- It covers the case where the runtime is a regular browser, a webview,

an OS such as Firefox OS or Chrome OS, or something like [node-webkit](#).

- The key point is that it's written for the technologies that are native to the web platform.
- The term makes a clear distinction from the serverside.
- It even works OK as an acronym if you're into that sort of thing.

I didn't coin the term, it was suggested by [Adam Brault](#) when I was complaining to him about this naming problem while, ironically enough, preparing for the talk I gave at [FFConf](#) called "A Single Page Story" in which I also discuss progressive enhancement, isomorphic apps, Twitter's famous switch back to server-rendered apps, etc. That talk is here, if you're curious:

Adam and I work together at [&yet](#) where we've been helping teams build Native Web Apps since before [Backbone.js](#) was a thing.

Anyway, the gist of that talk is that the web runtime has evolved into quite an impressive list of capabilities and with the addition of full offline support that [Service Worker](#) will bring, it becomes an incredible and very complete platform to build for. Just look at what it offers:

- HTML5, CSS3: for performant UI rendering (optimized for GPU control)
- Networking capabilities:
 - XMLHttpRequest: our HTTP client
 - WebSocket: for fast, bi-directional connections to servers
 - PeerConnection: for peer to peer networking
- WebAudio: for impressive audio creation and manipulation
- Data storage: indexDB, localStorage, sessionStorage
- WebGL/Canvas: for advanced rendering capabilities
- getUserMedia: for access to user microphone and video cameras
- Web Workers: for parallel processing
- Caching and Offline Support: via cache headers and app cache
- Geolocation
- Notification API

Coming soon:

- [Service Worker](#): a programmable cache that can intercept and choose what to do with network requests. In other words, incredibly precise control of offline behavior. It just landed in Chrome stable yesterday and [is in progress](#) in FF.
- Background processing and push notifications (also via Service Worker).
- [Standardized JSON App manifests](#) for describing installable web apps.
- [Web Payments](#)

Other benefits include:

- Dynamic loading of external libraries

- Zero configuration installs
- Expressive, extensible, event-driven native language
- These runtimes are already installed on billions of devices
- Can be "wrapped" into "native-feeling" apps on a wide variety of platforms
- It's easily the most open and accessible platform on the planet. We don't have to pay homage at the "walled gardens" of the world, like the Apple App Store.

I'd argue it's the most desirable and complete platform on the planet and I'm convinced we're just at the beginning of a long growth curve of these types of apps. If you're convinced otherwise or have other thoughts [let me know on Twitter](#).

Join us! Let's build amazing Native Web Apps and call them what they actually are.

You might also enjoy reading:

- [A microservices case study: Auth](#)
- [Lazymorphic Apps: Bringing back the static web](#)
- [Introducing JS for Teams, a training experience that's the first of its kind](#)
- [A microservices case study: Introducing Yeti Threads](#)
- [The web has outgrown the browser \(a web-lover's web-loving rant\).](#)