

Progressive web applications

Intro

When did you first **hear of PWA's** and when did you first come in touch with them. What were your initial thoughts.

I am an engineer specialized in Javascript. I have mainly been involved in Angular and back in the day Angular.js. I'm also part of the Google Developer Expert program. You can get this title by being part of the open source community and do a lot of talks. You get to try out the cutting edge technologies.

I heard about PWA's when Alex Russell wrote his infamous blogpost. In the beginning it was really experimental and it didn't have many features. I was playing with it in my own side projects, I wrote a lot of really bad manifests and service workers just to figure out how it works under the hood. But today progressive web apps are mainstream and production ready. So, so that's how I got to get to try out this new way of building applications. And that's how I discovered the technology.

Selling points

How and why would you convince your team to make your next application instead of a regular site or native app.

What are for you the main "selling" points of a PWA.

So, um, well, first of all, I don't like convincing people that's a bad thing to do. Ah, but however, I mean, if I need to convince someone, I would probably talk with the, with the business people because, having a progressive web app, I mean, yeah, I mean, it's cool. It's a new technology you get to play with around new stuff, but the real benefits of progressive web apps are. For business, mainly because, um, you get when you're right, a web app that is a progressive web app and like behaves and feels, like a native app. That's why what people are looking for.

For a company it's a huge benefit because it's cheaper to build, uh, cause like you don't have to build an Android app, an iOS app, a Windows app, ... You can just build like a web application with web technologies and then deploy it on a website. Uh, so, and it's easy also to update because you just update the website and everyone would have the new version.

Um, if you're building apps for mobile, uh, with Android and iOS. You go, you have to go through the store. So you have to pay fees and you have to pay Apple and maybe Google, whatever. Uh, whereas like when you build a JavaScript web application, it's on your server. You update whenever you want. You don't pay anyone.

It's also good for discoverability cause it's a web. Web page with SEO keywords and everything. So it's easy for people to find using Google search engine or Bing or Yahoo. So I would rather convince the business guys because it has more benefits for them than the engineering side of it.

Is not being in the app-store an advantage or a disadvantage?

you don't depend on apple or google to distribute the app. But you also lose a lot of traction by not being in a store.

Some people will think an app in the app store is more credible than a webapp

So, yeah, that's true. Because I mean, even for like indie developers who want to sell their applications on play store or on the Apple store, so it's not really great experience for them. Um, however, you can already start publishing your progressive web apps on the windows store and play store.

Apple is coming slowly to this, to this game, and they're, they're allowing people to, um, like a host or publish a progressive web app. doing this, you can like review apps, give them like a stars, whatever you want. Um. And, uh, even probably, I, I haven't, I haven't witnessed that. But you could call some mix and match, like native with progressive stuff. So you can also, all right, you can publish it on an app store and have some of the part paid. You know, you can pay for services and, but uh, I personally, I don't see any benefits of doing that either. You go progressive and get fully progressive web stuff. Or you can do it native because it's just going to complicate things on the engineering side if you combine them.

Previously in the past, if you want to install progressive web apps, you had to go to web a website and. Use the app for many, many times. Then you get the notification to install the application. Now you can download them from the first visit.

In what types of applications do you see PWA's excelling

the first category is eCommerce and, AliExpress is the best example of this. I think last time I, uh, I heard the news about that they, like, they did like plus 100% in revenues or three of the So, yeah, that definitely a great,

Other use cases are consumer based. for instance there is, there is a Twitter progressive web app., there is a Starbucks progressive web app. There is also Pinterest, these entertainment applications are also great examples. Because they they provide a service like you can consume a service. we can, yeah, we can imagine other, other categories around these like entertainment and music. I guess Spotify also has a progressive web app.

Maybe even things like communication applications like video calls, like Slack,... I don't know if they have a progressive web app at the moment, but they will at some point. I'm sure they will do it in the future. I mean, when you take Slack or teams, for instance, it's already an electron app. There is no reason for it not to be a progressive web app. They're already doing all the heavy lifting and heavy engineering. Um,

However, there are applications that's are not well suited for progressive web app are things like games for instance. Um, games are really, uh, like, uh, like heavy. I mean, they are, heavy in terms of resources, you know. it's better for them to be written in native C plus plus or these kinds of technologies instead of web. Because one of the characteristics of a progressive app is it needs to be smooth, you know, a smooth UI. Kind of animation and pleasant to use. And for games. I mean, of course, if you're building like, um, casual games, you know, that's, that's fine. But yeah, I mean, you cannot build Fortnite in as a progressive web app.

Adoption

- Some features that provide a better experience are **fairly easy** to implement. Why are most of the big websites **not making use of these features**? Why does facebook not have some basic caching which caches the application shell and provides some offline functionality?
- **react.js** makes it really easy to implement to convert a regular application to a offline capable PWA. But writing your own custom service worker is really cumbersome. Can you explain why this is not easier with a big framework as react?

So my opinion, i Don't know this but it might be politic reasons or internal reasons. They might not have any benefit of doing this because as you said, like on the engineering side, it's easy to do. I mean, anyone can do it. If they don't want to do it is theirs. For sure there is a political reason or financial reason for this. Do it's definitely not technical, but it's, I don't know, it's maybe business or political stuff, which is obviously not public and they won't say it.

It's probably not their priority right now. And Facebook is growing it started as like a social network, but now it's growing. Like it's, maybe it's a market. You can sell stuff on it. You can have chatbots, you can have lots, a lot of stuff. So maybe that's not compatible with progressive about what you're doing. I'm just guessing here. I don't really know the main reason.

Working in a big corporation, like I do right now. Most of these reasons are usually political and not technical. When it comes to politic, most of us, we don't know the reason.

Future of PWA's

One of the biggest reasons why windows on mobile didn't work was **convincing developers** to create applications for windows.

When Huawei couldn't use google services anymore this was probably also the biggest problem for huawei.

Both of these parties have resources enough to develop their own OS. But convincing developers to develop for their OS is not that easy.

At the moment there is no doubt that Google is pushing the PWA movement the most. And it's really 'googly' thing to do, but I don't really understand why they are doing this.

Isn't **PWA's a thread to google**, because this could mean to losing their oligopoly with Apple.

How do you see this this evolving for other parties and do you think that google will keep on innovating on PWA's?

So one thing I learned, during this all this time, talking with, uh, Google engineers or working with them on some open source projects or all the things I've heard, like I'm not, I'm not working for Google obviously, but the things I've heard like, And it's the same thing for Microsoft. So, uh, one thing to learn to know is that like Google and Microsoft and other big corporations, have many engineering teams with different specialties, for instance, there's an Android team, the Chrome team, the angular team, the GCP, ... And every organization has their own priorities, their own budgets, their own ... , it's like a small company inside the company .

Google is not making money of Android. Google is an ad search company. So they are making big money on that, So having an operating system like Android, just help them push more ads to people. That's why Android is an open source project and parts of the OSP organization and everything, uh, they're also making money from GCP, cloud. So that's why it's not open source.

Having the Chrome team, it's not Google, but the Chrome CR pushing progressive web apps. And of course from the outside, you don't see the Chrome team, the underlying team. You see just Google the Google brand. It, same thing for Microsoft. If you have, like, if you have like the TypeScript team pushing TypeScript. Whereas like Microsoft, they have like C# and all the other technologies. Uh, it's like small teams are pushing hard, their work. and from the outside just you just see the peer corporation. So coming back to your question, I don't think it's there is any conflict here between progressive web apps on Android. They have the biggest market share, you know, it's open source. Anyone can take it and build their own, uh, as you said. So having the Chrome team pushing progressive web apps and especially web in general. So that's the main purpose behind this progressive web app set of technologies. By pushing progressive web apps, like indirectly, they're also pushing products like Chrome, like other things like specifications they're working on,,for instance, the web USB they are working on with many Googlers and Google engineers. Uh, of course there are other, like there, there is like a Samsung internet there, Microsoft, some of them are also pushed by Apple. So it's also a way for them to push other that services and I don't personally see any conflict in here between Android as an operating system and progressive web apps.

Installation of PWA's

We, as 'tech-savvy' people know that if an **installation prompt** is fired, that it is a pwa and nothing harmful.

The problem though is that most non 'tech-savvy' people just continue using the webapp without installing it.

probably also not using the features that a pwa can provide

- example: **they won't use a PWA offline**, because they don't know that it's capable of doing that.

Do you have a solution for this and do you expect a shift in user behavior?

If the users sees that message on the bottom And they can simply click on the close and they won't see it ever. Um, so that shouldn't be a problem. Most of them, maybe they would just see the message and ignore it I guess. Or if they are coming to a website to read something or do something on the website it's like when you visit a site and you have ads, swerving at your face, and you just click and say, okay, leave me alone. I think they would do the same thing with the install banner, uh, at the bottom. Uh, but yeah, for the future. I don't have any information about that if, if this is going to evolve, but. One thing. I'm definitely sure about that, if there is any problem related to this feature, I'm sure. specification would evolve too, to deal with this new kind of, of issue. That's one of the benefits of working on the web. I don't see any issue with that because many people would just ignore the message message

For a company like AliExpress or like a big company. If they want to add this kind of feature, they probably do a lot of specific research that this feature will be used by the consumers.

And in the end, it's really easy to make a website offline capable, so it's not a big investment to do that.

Apple

Apple is as often not that keen on giving access to their OS. This is the case for PWA's. **Do you expect better support** for IOS and macOS and safari?

I don't know, i just use Apple products, but i don't know anything about Apple.

However, I was going through Twitter recently. I saw that they hired a lot of people I know. Most of them are joining the open source organization inside Apple. So maybe there is a message hidden. They're there. They're probably starting to build a open source community around the products they're building. But, but that's, that's the only thing I can guess from what's happening. UYou should probably ask someone who is really familiar with Apple, uh, like from the beginning.

Look how the web specs works, it's like aevery big company, every company has a chair into the web spec. And they all can have like their veto. So if Google for instance tells, okay, I have this new API I want to push, I want to be, make it public and everything. so Microsoft can say, okay, I will build it in edge. But if Apple says, I won't do it in, in Safari. You can do anything you want, they won't do it. And that's it. That's end of story. If you can convince them, you can. if you can't you cant

But I've been following some of the meetings of the W3C. There is a lot of discussion going on and some of them are really public, you can probably check them out.

But to conclude the question, it's all about politics

Beginning of pwa's

you have a nice amount of experience, **what are the biggest shifts you have experienced** on the web.

- Do you think that pwa's will stay a 'niche'? Or do you think it will become the mainstream for webapplications to register a service worker and add service worker functionality?

So, I'm witnessing a lot of all the work that, um, not even Google, but also other engineers, even Microsoft, some of people at Apple and all the major companies and also other companies like Samsung, like Intel, they're all doing a lot of work around the web and the underlying layers of the web.

for instance, eh, so if you want to define what's the progressive web app, it's an app that behaves and then feels like a native app. This. When you think about a native application, i has access to camera, has access to a microphone, it has access to all the sensors, to your, your contacts or SMS, everything. I mean, it's a native app. You have access to the whole OS.

Intel is, has been working on an WEB NFC. There's also web Bluetooth, there's a lot for the. I think there is a new standard. It's called, um. Web of things. it's like an umbrella spec around all this hardware access.

A lot of companies are doing really great job for this, and I can see that progressive web app to benefit from that. Obviously they're doing this to expose API to javascript and on the web in

general.

So I think even like in , maybe three, four coming years. We'll have even more features for progressive web apps. I mean, today we have lots of stuff.

So my prediction is progressive web apps are, are, are here to stay. I'm talking about the technology and not the politics behind it. These will probably change.

I even think that in the future you will be able to monetize you progressive web app. Like we do today with native apps. It's becoming a big part of the web.

PWA's is not one API or one technology, it will probably evolve. And that's why I believe PWA's are here to stay

Anytime a feature is added to the web, it's there forever. It's impossible to remove, because then a lot of websites would brake. So now they can't stop supporting service workers and service worker functionalities.