Progressive Web Apps

Web Apps with super powers

About Me

Head of FE + JS @ CodeSparks

codeSparks
by AgileSparks




@giladaya

So... we need an app

Traditionally:

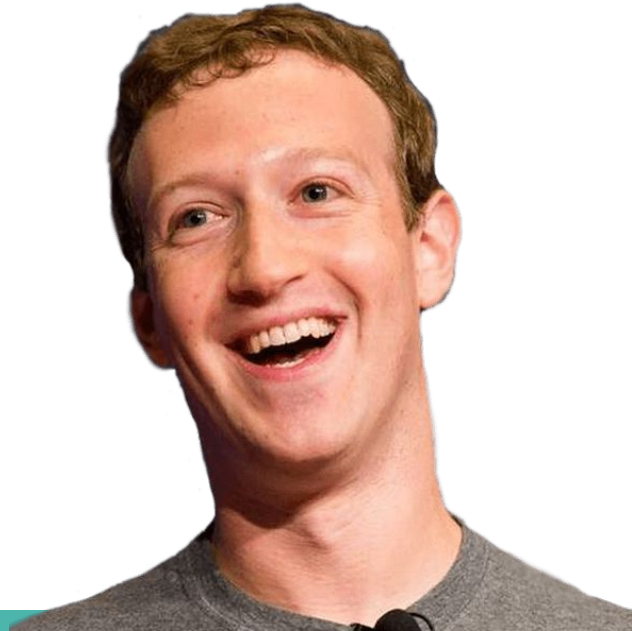
1 x Android app

1 x iOS app

1 x Web app

But... resources are limited

To most at least...



What if I told you

You may not need native apps!

~~1 x Android app~~

~~1 x iOS app~~

1 x Web app

The advantages are obvious!

Wait, isn't the UX bad?

There's a big difference between a webapp and a native app!

Not necessarily

Looking for the holy grail

A universal app technology

Examples:

- Java / Swing / JFX
- Phonegap / Cordova
- Ionic
- React Native (now with Windows support)
- Flutter (now with web support)
- PWAs

A bit of history

2007: webapps for the iPhone

2009: Chrome OS

2013: Firefox OS

2015: PWA from Google

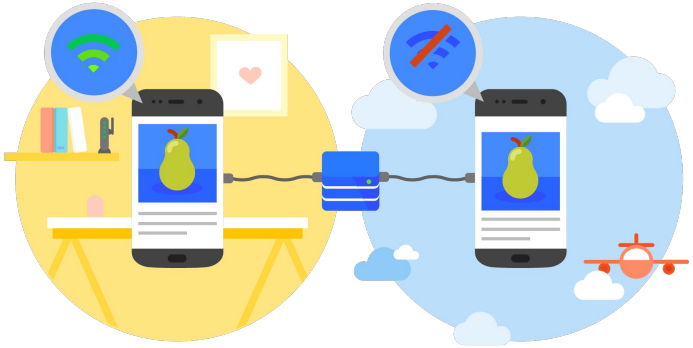
2018: Support in Safari

What are PWAs?

As Google defines them:

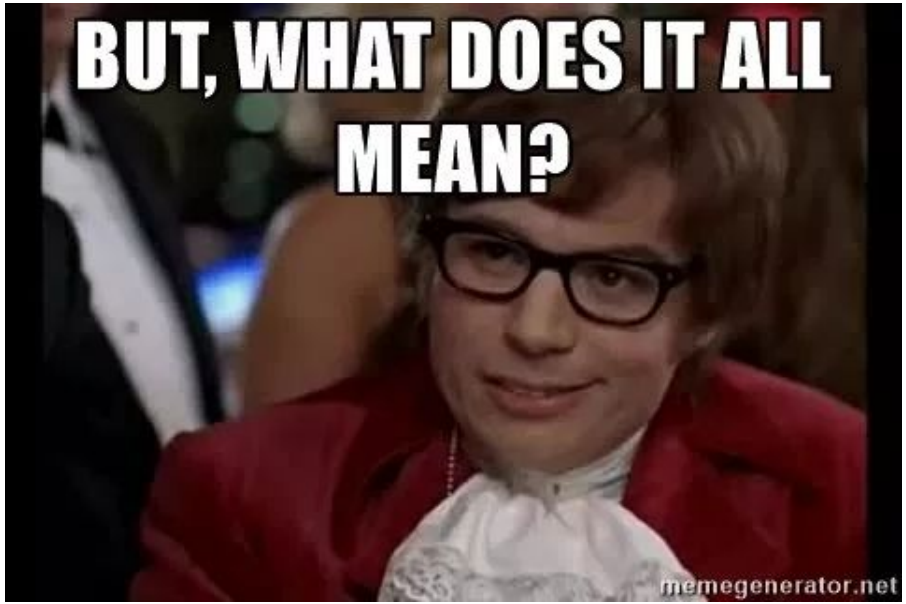
Progressive Web Apps are user experiences that have the reach of the web, and are:

- Reliable
- Fast
- Engaging



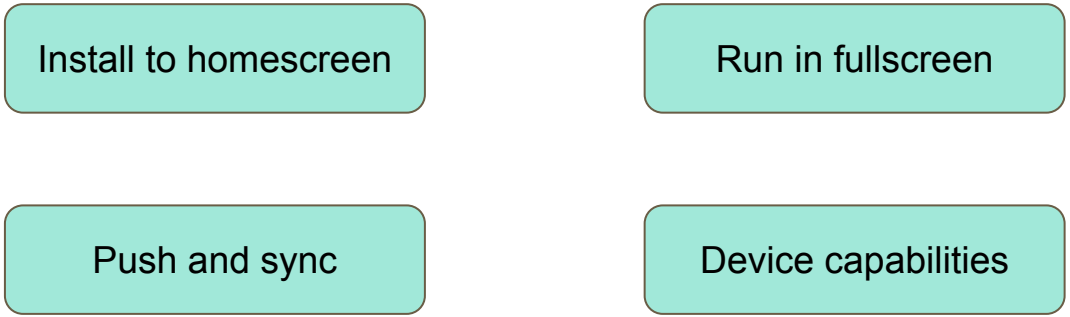
What does it mean?

Basically - web apps that behave like native apps



What does that mean?

- **Reliable** = work offline
- **Fast** = local cache, focus on performance
- **Engaging** = native-like features



Demo Time

Let's make a PWA

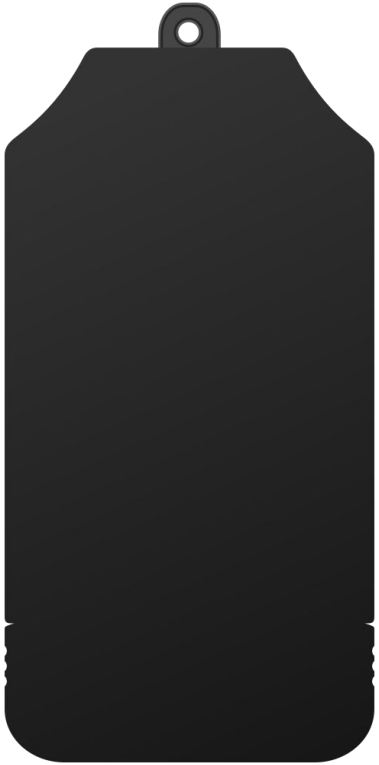
In three easy steps

Step 1: A normal web app

Must: Served over HTTPS

Should: Mobile friendly

Any guesses?



Let's see it!

Repo

Web APIs

Storage

Audio and video

Sensors

Camera

Bluetooth, USB

Location

Notifications

GPU

Vibration

Speech

Virtual Reality

Sockets

...

Aside: Have Web Standards on Mobile Caught Up to Phonegap in 2017

Are we there yet?

The lighthouse will guide us!

Step 2: The manifest

JSON file

Info for installing to homescreen

Use a generator

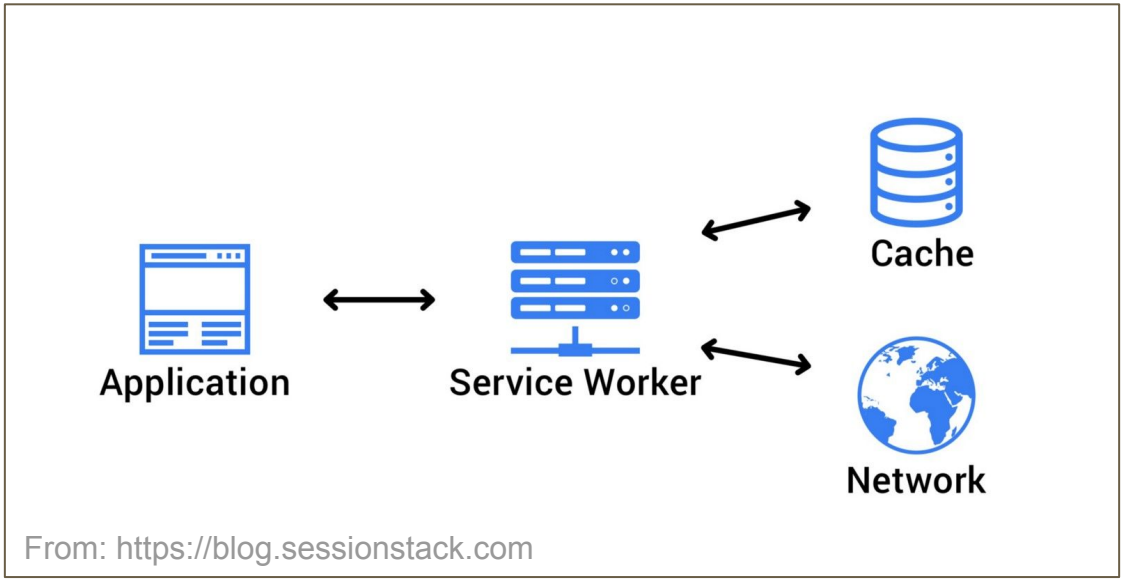
Are we there yet?

Let’s check!

Step 3: The service worker

Programmable network proxy

Only run over HTTPS



From: https://blog.sessionstack.com

Service worker capabilities

Cache

Push notifications

Background sync

Service worker pitfalls

There are 2 hard problems in computer science:

1. Cache invalidation
2. Naming things
3. Off-by-1 errors

(Paraphrase on a quote by Phil Karlton)

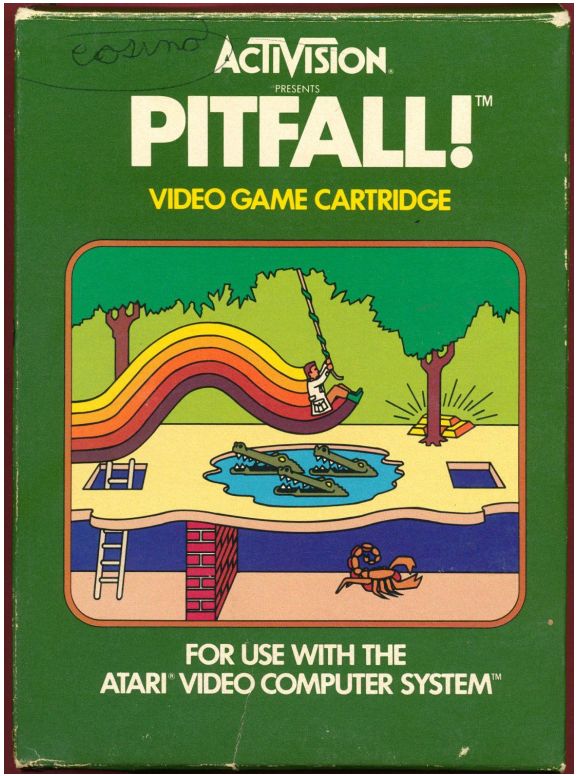
Service worker pitfalls

A cache

A cached cache

Low-level

Can hurt performance



Service Workers lifecycle

Service worker

Gotta add it

Are we there yet?

The desktop experience

No install prompt 😕

But works offline! 🙂

Meaning of offline

Challenges

Inbound / Outbound

Assets

Data

Solutions

Cache

Canned response

Background sync

Offline mode

Now on mobile...

Try it yourself :)

https://perpetual-join.surge.sh

Now on mobile...

Now on mobile...

How it works?

Chrome generates and installs an APK

Registers some intent filters

Shared storage with browser

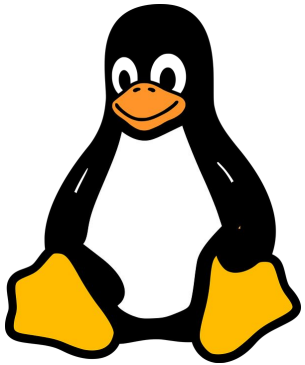
Does not use the WebView

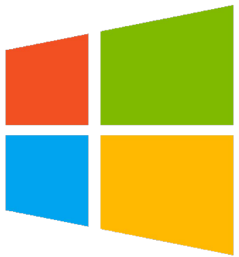
More info

Can we do something for desktop?

Yes!

From Chrome 73 (March 12, 2019) desktop PWAs are supported on all desktop platforms!





Challenges

Why not?

Limited support / compatibility

Complexity of solution

Limitations on apps

Distribution

Complexity

Use workbox

Use appshell pattern

Limitations

https://giladaya.github.io/whipped/

Distribution

As a web app

Monetization is an issue

Publish in the Play store (complicated)

Try PWABuilder

Still...

Many suitable applications and use cases

Also beneficial for standard webapps

Case Studies

Flipcart Lite

One of the first

Soltell

Self-funded, part-time venture

PWA enabled quick time to market

Challenges with iOS and Windows

Pinterest

3 month to implement (Jul 2017)

Weekly active users on mobile web: +103% (+312% in India)

Session length: +296%

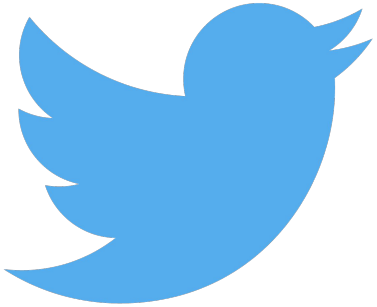
Logins: +370%

New Signups: +843% yoy

800k weekly users of PWA as native in < six months

Sources: https://medium.com/@Pinterest_Engineering/a-one-year-pwa-retrospective-f4a2f4129e05
https://medium.com/dev-channel/a-pinterest-progressive-web-app-performance-case-study-3bd6ed2e6154

Twitter



Twitter Lite available April 2017

Pages per session: +65%

Tweets sent: +75%

Bounce rate: -20%

250k unique daily users of the app from homescreen x4

Source: https://developers.google.com/web/showcase/2017/twitter

Trivago

Repeat visits: +150% (for I2HS users)

Clickouts to offers: +97%

500k installs to homescreen

Source:
https://www.thinkwithgoogle.com/intl/en-154/insights-inspiration/case-studies/trivago-embrace-progressive-web-apps-as-the-future-of-mobile/

What's next?

Additional platforms

Badging for the launch icon

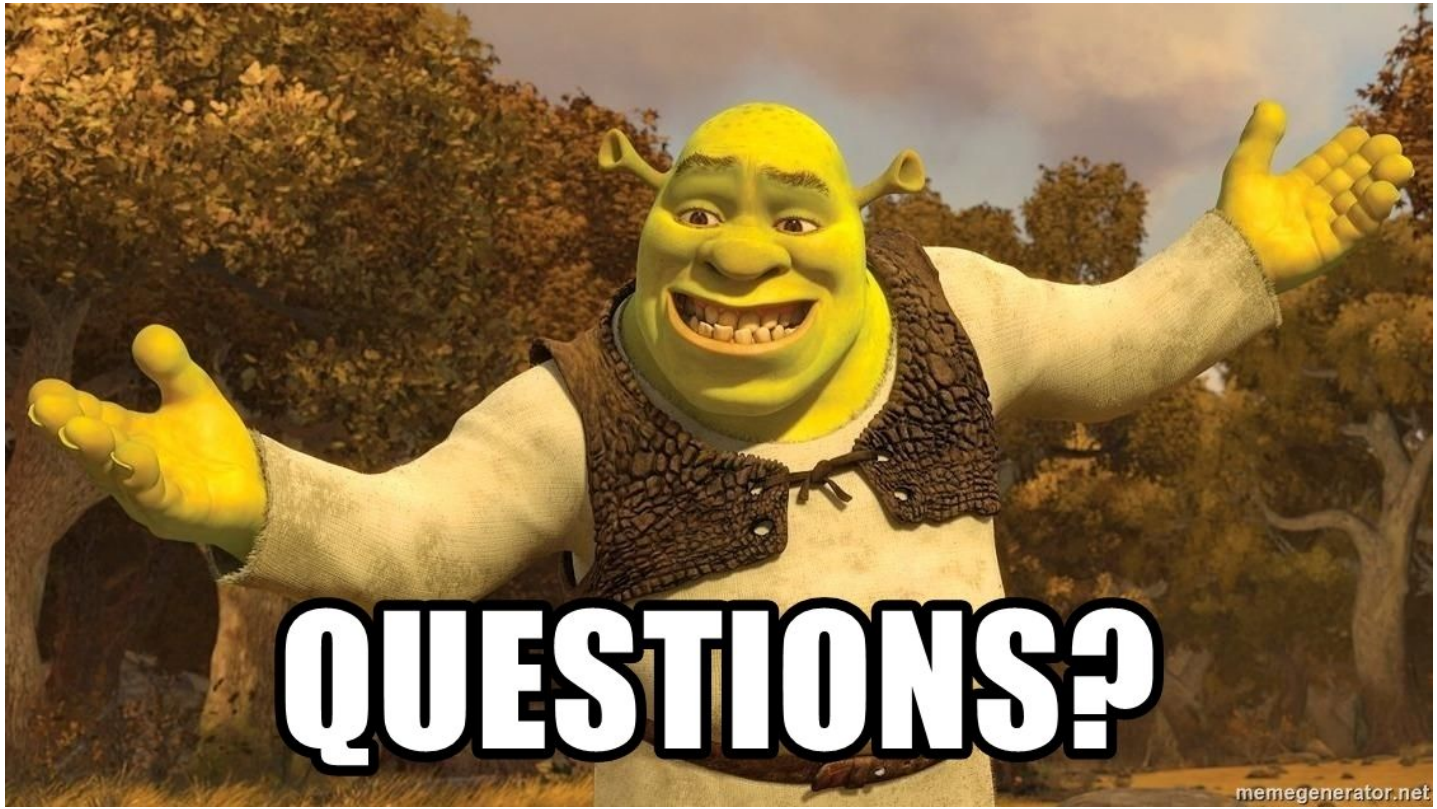
Link capturing

Omnibox badging

Keyboard shortcuts

Summary

- PWAs: web apps with native-like capabilities
- Easy to convert existing web apps
- Mainly: add manifest and service worker
- Service workers are powerful but complex
- Use tools to test compatibility and generate
- Support is pretty good and growing


QUESTIONS?
memegenerator.net