

Progressive Web Applications: Blending Industries

Obligatory "Who Is This Guy?"



Jack Pritchard
@[WhatJackHasMade](#)

Freelance Front-End Developer

Blend, Merge, Connect

With the rise in use of mobile devices, there has been a clear divide between native mobile applications and online websites.

Aims of this Presentation

This presentation aims to inspire developers to improve their applications with cutting-edge technologies.

Challenging industries and challenging the decision to split into sub-segments within software development.

Before We Begin

Let's Define Our Industries

Online Websites and Browser Applications

Usually consisting of front-end languages such as HTML, CSS and JavaScript.

With a back-end system built on PHP, .NET, or other serverside language.

iOS, Android and Other Native Mobile Applications

Seperate from websites, these applications are often coded in languages customary to the device.

Java is commonly used for Android development and Swift for iOS.

Defined Industries

Browser Applications vs Mobile Native Applications

Why Do Mobile Applications Exist?

Why Do Mobile Applications Exist?

Mobile applications solve problems, providing a reliable solution for creating applications with capabilities to store large amounts of data and files offline.

Most mobile applications can be used without an internet connection.

Not to mention the access to inputs from your device such as microphones, cameras, and motion sensors.

New Problems, Old Solutions

The first commercially successful smartphone, the iPhone was released in 2007.

More than a decade later, web technologies have come a long way.

Challenging the need for native applications to solve new problems.

So What Can the Web Do in 2018?

Accessible Inputs To The Browser?

- Camera ?
- Microphone ?
- Orientation/Tilt ?
- GPS ?

HTML & JS Webcam

```
navigator.getUserMedia({ audio: false, video: true },  
    trueFunction, falseFunction);
```

HTML & JS Microphone





```
navigator.getUserMedia({ audio: true }, trueFunction,  
                        falseFunction);
```

You must allow your microphone.

Device Orientation



Inputs

- Camera 
- Microphone 
- Orientation/Tilt 
- GPS 

So We've Got Our Inputs

What Solutions Do Browsers Have For Storage?

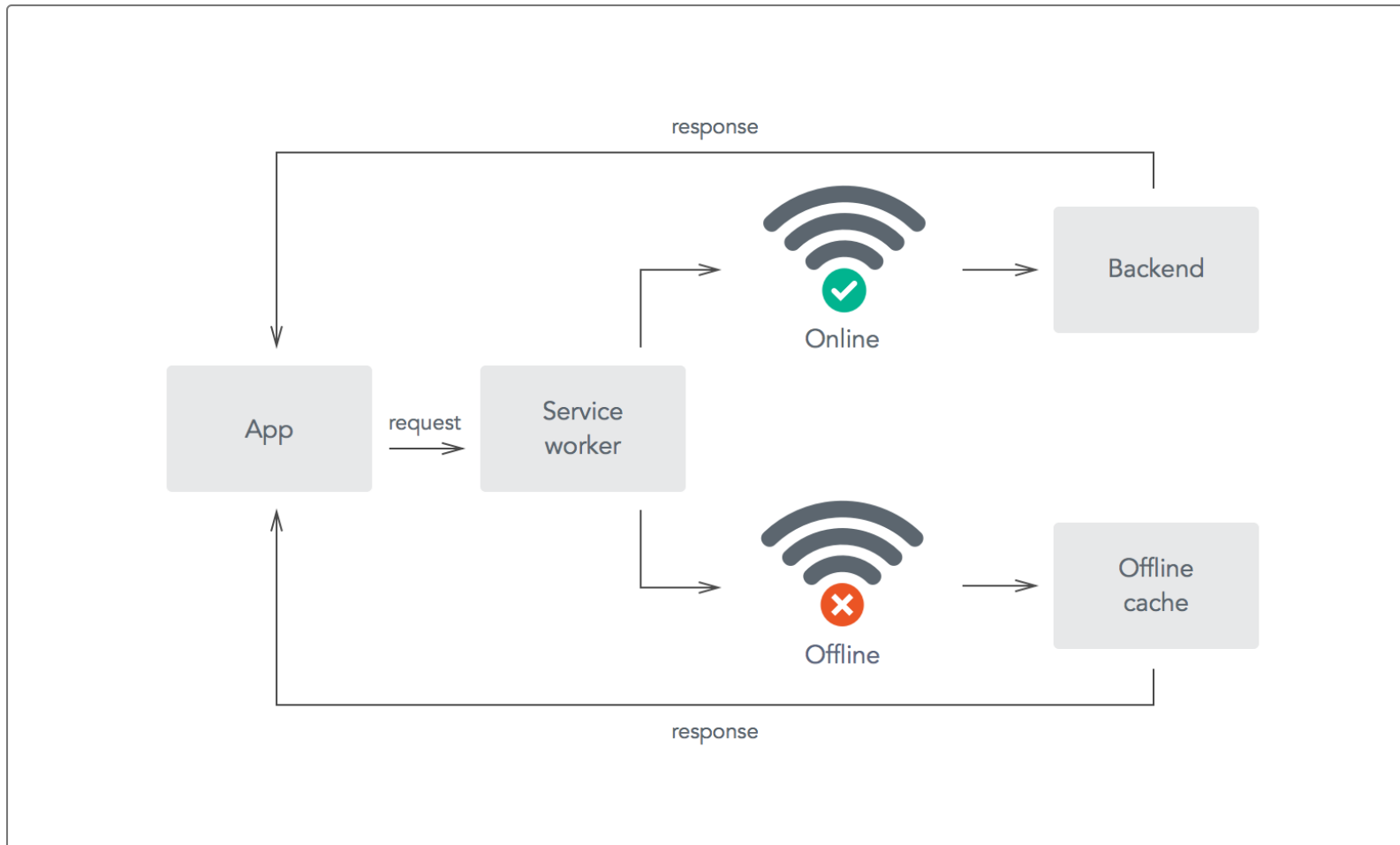
Files - Service Workers

A service worker's primary function is to intercept HTTP/HTTPS calls made by pages.

Service workers essentially act as proxy servers that sit between web applications, and the browser and network (when available).

Essentially it's client-side caching.

Files - Service Workers



Source: <https://pjagajitprusty.github.io/PWD/>

Data Storage - IndexedDB

IndexedDB is an API that allows developers to store or locate records in the browser directly.

Current State of The Web

We are now entering an exciting time for the world wide web.

With access to inputs otherwise reserved for software, and the ability to store large amounts of data or files for offline use, we are now given the power to replicate the power of offline tools.

And So Enters PWA

Progressive Web Applications

Imagine a website that works offline and online.

A website that can be accessed in-browser or as if it were a native app,
anywhere in the world, anytime.

Progressive Web Applications

Progressive Web Applications leverage the technologies we've covered.

Embracing the concept of offline storage of data and files.

Accessible just as they would be any other piece of software.

Progressive Web Applications

"A Progressive Web App (PWA) is a web app that uses modern web capabilities to deliver an app-like experience to users." - ionicframework

The term PWA, coined by Google has seen a rise in popularity due to an increase in browser support for the essential technologies PWAs use.

Visiting Google



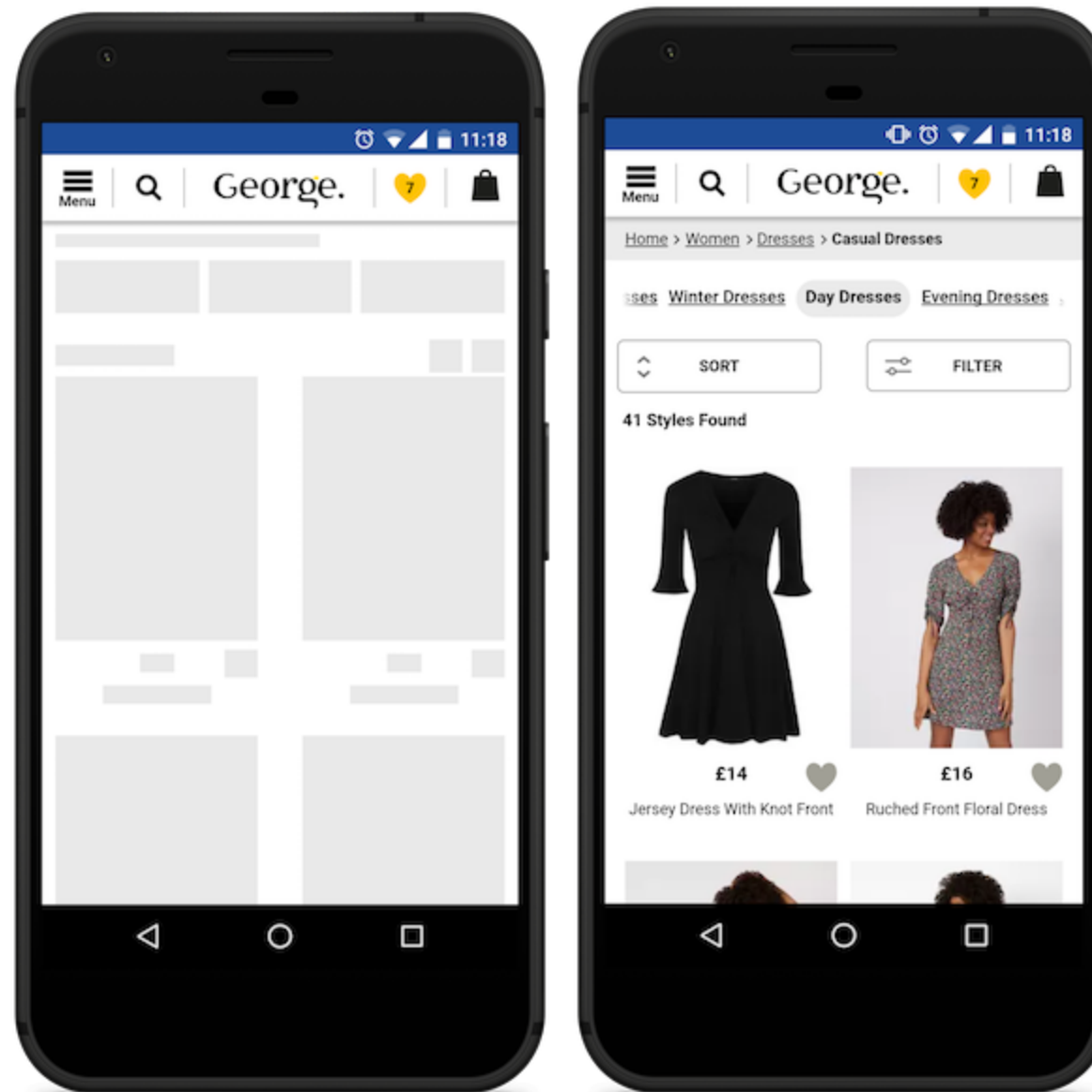
Fortunately I was given the opportunity to attend Google's team training days on Progressive Web Applications at their offices in London, back in 2017.

Main Takeaways

- Progressive Web Applications aren't a new trend.
- They're a new concept that build on existing technologies.
- Caching information and presenting websites to users in a familiar design.
- A website can benefit from PWA technologies, even if not a complete PWA.

Enough Talk, Let's See Some PWAs

A PWA Case Study - George.com

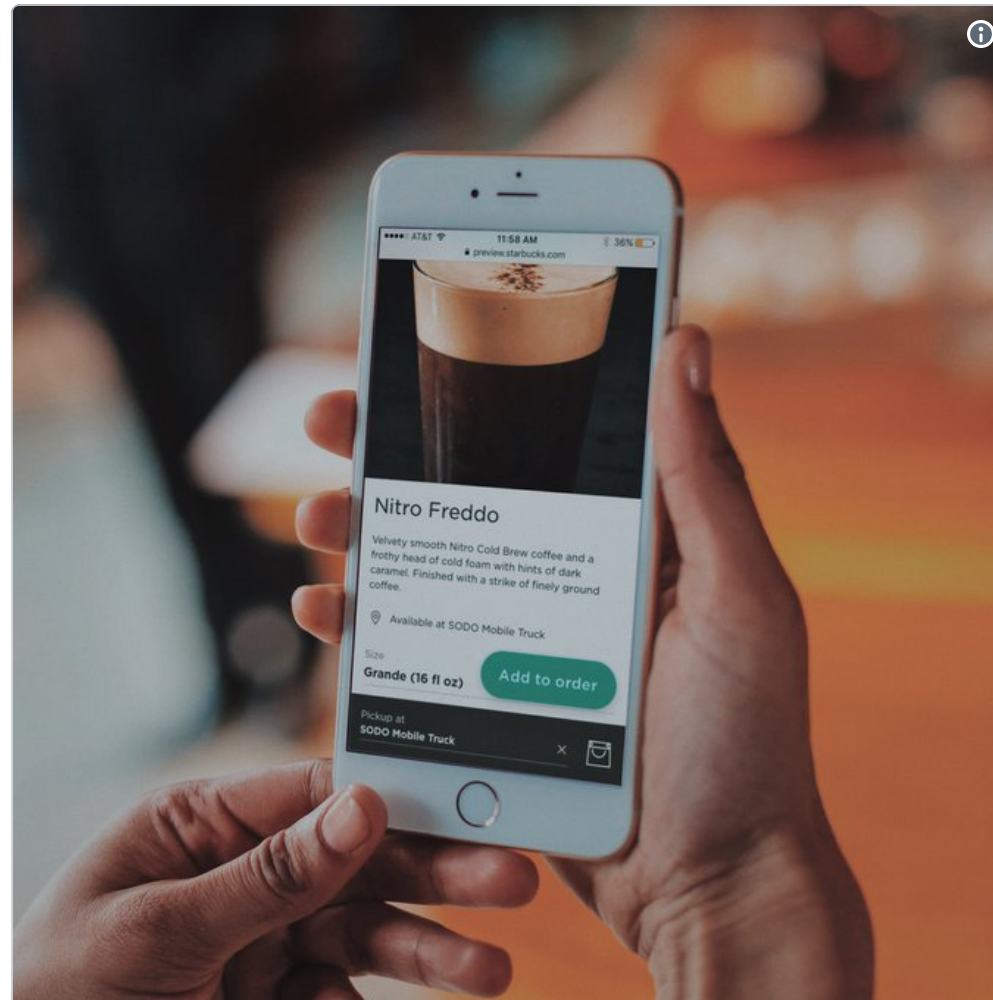


Familiar Design?

A PWA Case Study - George.com

- 3.8x - Faster average page load time
- 2x - Lower bounce rate
- 31% - Increase in Conversion Rate
- 20% - More page views per visit
- 28% - Longer average time on site for visits from Home screen

PWA Case Study - app.starbucks.com



David Brunelle
@davidbrunelle



My team at [@Starbucks](https://www.starbucks.com) has been building a PWA, and it's now in beta! Check it out at preview.starbucks.com if you're an existing customer!

♡ 461 11:13 PM - Sep 7, 2017

💬 186 people are talking about this



PWA Case Study - app.starbucks.com

Starbucks iOS: 146MB // Starbucks PWA: ~600KB

It's 0.4% of the size.

The PWA is less than 1/243 the size of the Starbucks iOS application.

Then, of course on iOS it then also still has to install and boot up!

Cool, Let's Make Our Own

Expelled Skateboarding

Expelled Skateboarding Demo



Before We Dive In

What needs be implemented to be considered a PWA?

- Registers a service worker
- Responds with a 200 when offline
- Contains some content when JavaScript is not available
- Uses HTTPS
- Redirects HTTP traffic to HTTPS
- The user can be prompted to install the web app
- Configured for a custom splash screen
- Address bar matches brand colours
- Has a `<meta name="viewport">` tag with width or initial-scale
- Content is sized correctly for the viewport
- Web Application on Android Homescreen

How Are We Analysing The Project?

Google Chrome is host to numerous developer tools.

One of which is the ability to audit a webpage.

Registers a service worker

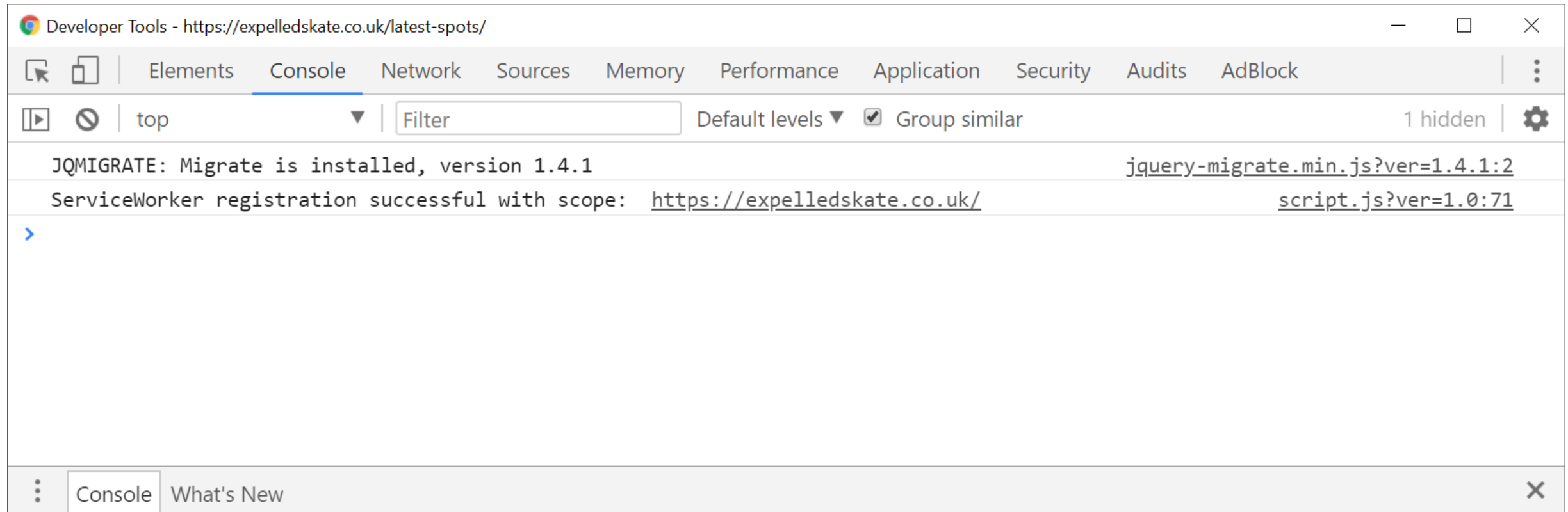
Service workers act as a cache for the storage of web files in the browser, preventing the need for files to be downloaded several times on a site.

Service workers are enabled through the use of Javascript and will cache files in the directories specified.

Registers a service worker

```
'serviceWorker' in navigator) {  
  window.addEventListener('load', function() {  
  
    navigator.serviceWorker.register('https://expelledskate.co.uk/sw.js').then(function(registration) {  
      console.log('ServiceWorker registration successful with scope: ',  
registration.scope);  
    }, function(err) {  
      console.log('ServiceWorker registration failed: ', err);  
    });  
  });  
}
```


Registers a service worker



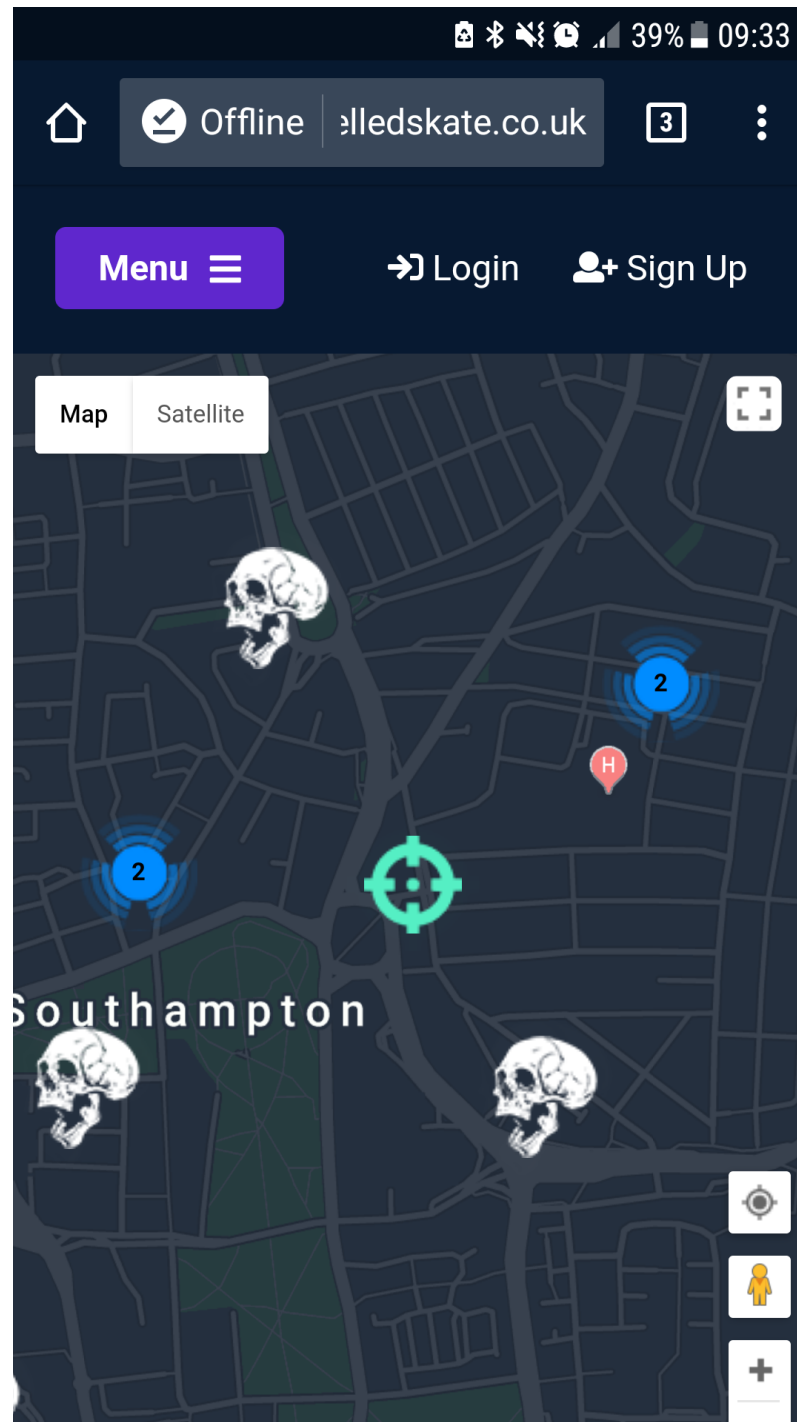
A service worker has been registered, this service worker is storing assets such as images, stylesheets, scripts and HTML files for offline use and to increase performance online.

Responds with a 200 when offline

When navigated to without an internet connection, the site needs to respond with a 200 HTTP code.

Provide a fallback offline `.html` alternative if the site is built on Server side rendering languages.

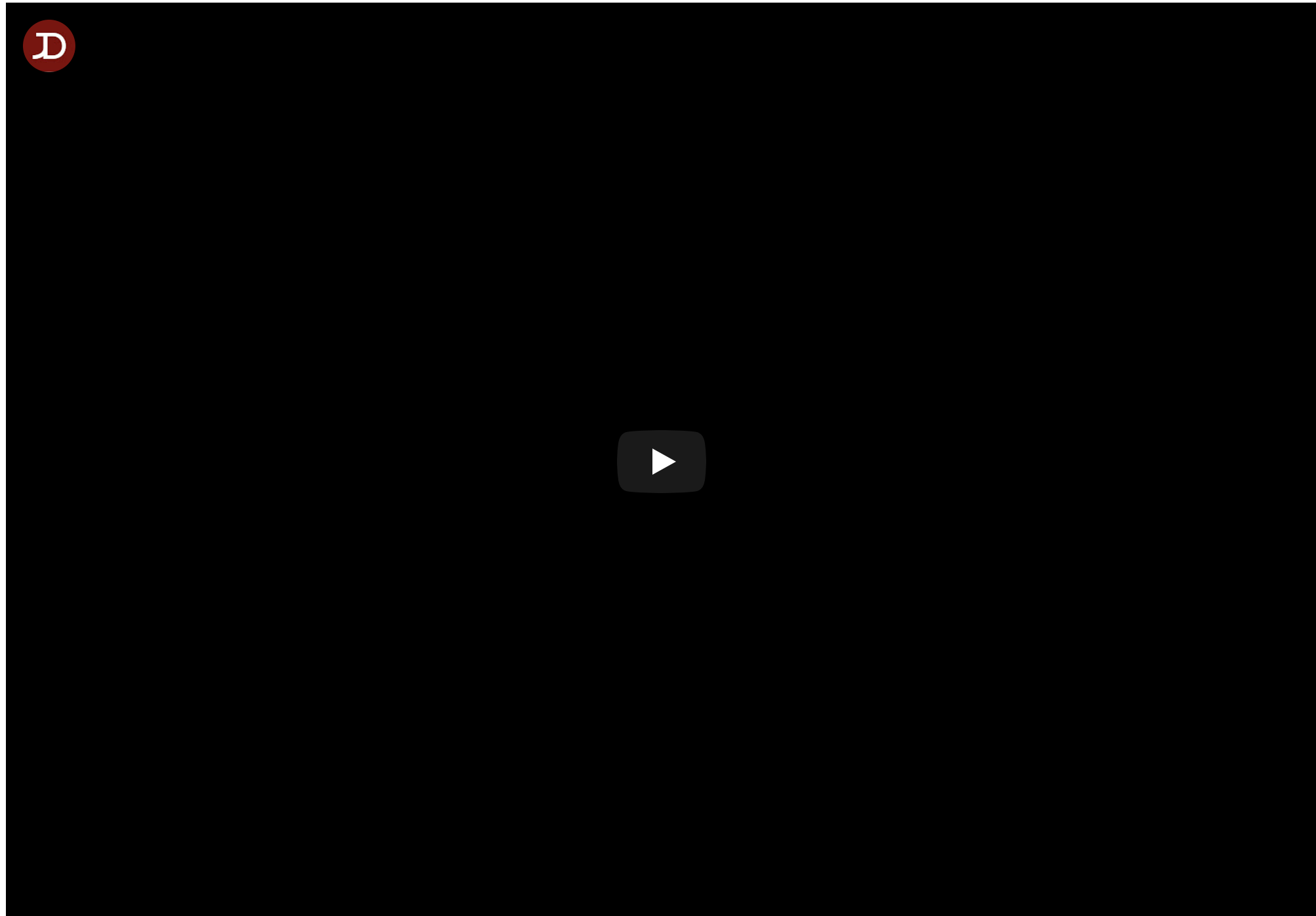
Responds with a 200 when offline



Contains some content when JavaScript is not available

When JavaScript is disabled in-browser, the following requests are made and complete when accessing the homepage of the Expelled Skateboarding site.

Contains some content when JavaScript is not available



Uses HTTPS

The Expelled Skateboarding site is hosted by DigitalOcean on a Virtual Private Server running Ubuntu 16.04.

The server uses Apache to configure the domain routing.

An SSL certificate was generated for the domain <https://expelledskate.co.uk> and added to the Apache configuration.

Redirects HTTP traffic to HTTPS

There are no areas of the site that request HTTP protocols internally.

```
RewriteEngine On  
RewriteCond %{HTTPS} off  
RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
```

To ensure all requests on the site are made using HTTPS, Apache has been configured accordingly.

The user can be prompted to install the web app

Users of the Expelled Skateboarding website may be prompted on mobile devices to notify them of the option to install the web app on their device.

The minimum requirement for prompting of installation of a web app is a web manifest file.

http://expelledskate.co.uk/wp-content/themes/expelled_theme/img/favicons/site.webmanifest

Configured for a custom splash screen

A custom splash screen has been configured in the manifest file.

Icons are set at 192px by 192px, 384px by 384px and 512px by 512px.

Configured for a custom splash screen

```
icons: [  
  {  
    src: "https://expelledskate.co.uk/wp-  
content/themes/expelled_theme/img/favicons/android-chrome-192x192.png",  
    sizes: "192x192",  
    type: "image/png"  
  },  
  {  
    src: "https://expelledskate.co.uk/wp-  
content/themes/expelled_theme/img/favicons/android-chrome-384x384.png",  
    sizes: "384x384",  
    type: "image/png"  
  },  
  {  
    src: "https://expelledskate.co.uk/wp-  
content/themes/expelled_theme/img/favicons/android-chrome-512x512.png",  
    sizes: "512x512",  
    type: "image/png"  
  }  
]
```

Address bar matches brand colours

Again, the address bar has been set to match the brand colours, setting both the `background_color` and `theme_color` property to `#071930`.

```
{  
  "background_color": "#071930",  
  "theme_color": "#071930",  
}
```

Has a `<meta name="viewport">` tag with width or initial-scale

The head tag of each page includes favicons, title, and meta information.

This is where the meta name viewport is defined to scale at 1.0.

```
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Content is sized correctly for the viewport

The Expelled website is built mobile first.

Images and videos resize, no content overflows on the page.

Web Application on Android Homescreen

Including a Web Application on Android Homescreen, introduces users to concepts they already understand.

This includes shortcuts on home screens, that will help blur the line between a native mobile app and a progressive web app.

This creates a smooth transition to PWA implementations.

PWA Complete



Recap

- Registers a service worker ✓
- Responds with a 200 when offline ✓
- Contains some content when JavaScript is not available ✓
- Uses HTTPS ✓
- Redirects HTTP traffic to HTTPS ✓
- The user can be prompted to install the web app ✓
- Configured for a custom splash screen ✓
- Address bar matches brand colours ✓
- Has a `<meta name="viewport">` tag with width or initial-scale ✓
- Content is sized correctly for the viewport ✓
- Web Application on Android Homescreen ✓

Are PWAs Compatible? - Service Workers

is
SERVICEWORKER
ready?

Yes.

[Status](#) * [Spec](#) * [Intro](#) * [Resources](#) * [GitHub](#)



Are PWAs Compatible? - Web Manifest Files

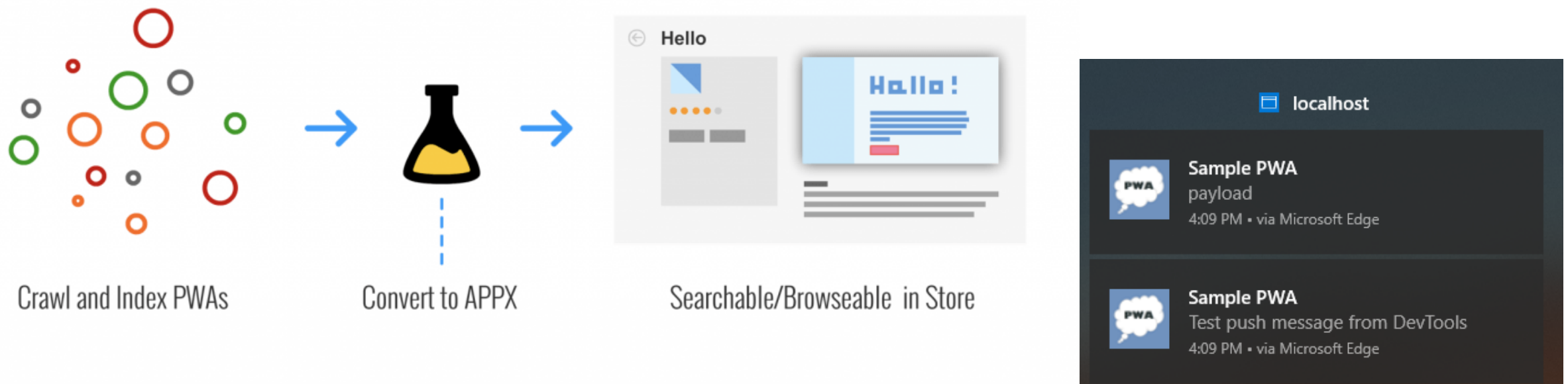
<https://caniuse.com/#feat=web-app-manifest>

New Support - Apple

Since iOS 11.3, Apple now has support for service workers and progressive web application technologies.

This update in support increased the scope of users that are able to take advantage of progressive website application technologies (service worker, manifest files, etc.).

New Support - Microsoft



Not only are website developers creating PWAs for the world wide web, mobile phones, but also desktop operating systems.

PWA Rocks

PROGRESSIVE WEB APPS

all

business

demo

event

game

news

reference

shopping

social

tool

other



WEBNFC SHOPPING CART



AIR HORNER



PAPER PLANES



SELIO

RioRun

RIORUN



KONGA EZ



Further Learning

- Henrik Joreteg - <https://joreteg.com/blog/betting-on-the-web>
- Google Case Studies - <https://developers.google.com/web/showcase/>
- Jeremy Keith - <https://abookapart.com/products/going-offline>
- Microsoft PWAs - <https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps/get-started>
- PWA Stats - <https://www.pwastats.com/>
- React PWA Tutorial - <https://medium.com/front-end-hacking/build-a-realtime-pwa-with-react-99e7b0fd3270>

Where to Use PWAs?

PWAs are beneficial to the majority of websites.

No user is going to suffer from cached files, unless files are constantly changing on a website.

In which case, if approached correctly, service workers can invalidate the cache and present new files.

Where not to Use PWAs?

- Large applications such as mobile video games
- Existing client projects - Sometimes large refactoring
- With clients that don't share an understanding on the benefits of PWAs

Bonus Slide: Style like a native application



Full featured HTML framework
for building iOS & Android apps

Download

Get Started

Docs

Templates

Apps Showcase

Forum



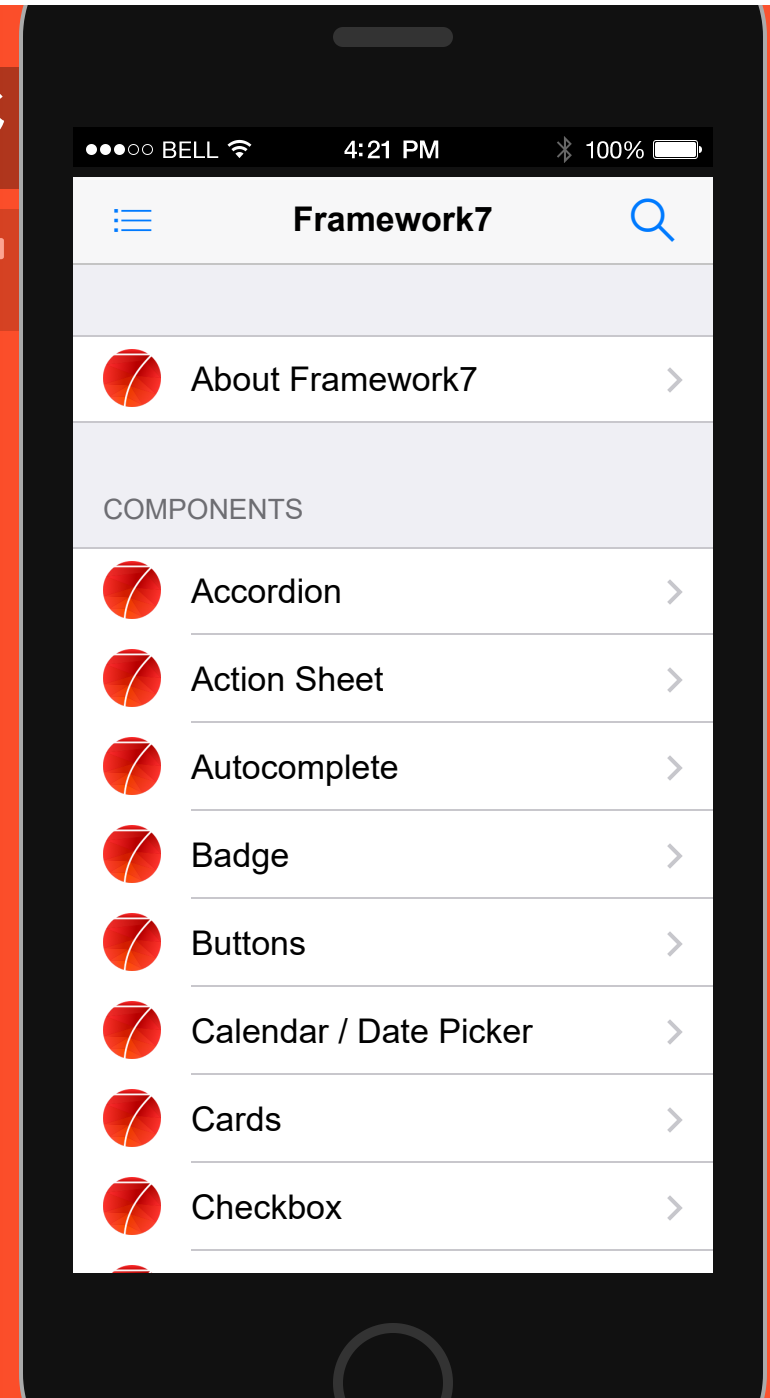
Get WebStorm, the
smartest JavaScript IDE.
Free trial, license from \$77.

ads via Carbon

MIT Licensed, v3.5.2 released on November 12, 2018

 13,389 stars 2,838 forks

 Support Framework7



Conclusion

Hopefully this talk has given you some food for thought.

I haven't presented a step-by-step tutorial on building a PWA.

I have simply touched on each requirement necessary to build a PWA in 2018.

Conclusion

If you, like myself, are excited by what these new concepts and technologies allow developers to create, then I suggest further research into how you can implement a service worker into your sites today.

Everyone benefits from them, even sites that don't have offline functionality.



Any Questions?

Jack Pritchard

@whatjackhasmade