

# Progressive Web Apps

A wide-angle photograph of a modern office space. Large floor-to-ceiling windows on the left provide a view of a city skyline. The office has a rustic feel with light-colored wooden desks and chairs. Several people are working at their desks, each equipped with multiple computer monitors. A wall on the right is covered in a grid of small framed pictures or logos. The overall atmosphere is bright and collaborative.

# Timon van Spronsen

## Front-end developer

**Wat is een  
Progressive  
Web App?**

**Progressive enhancement + moderne API's**  
**= cross-platform "native-achtige"**  
**applicaties**

# Progressive enhancement

- Werkt voor elke gebruiker
- Extra functionaliteit voor moderne browsers
- Polyfills

# Moderne web API's

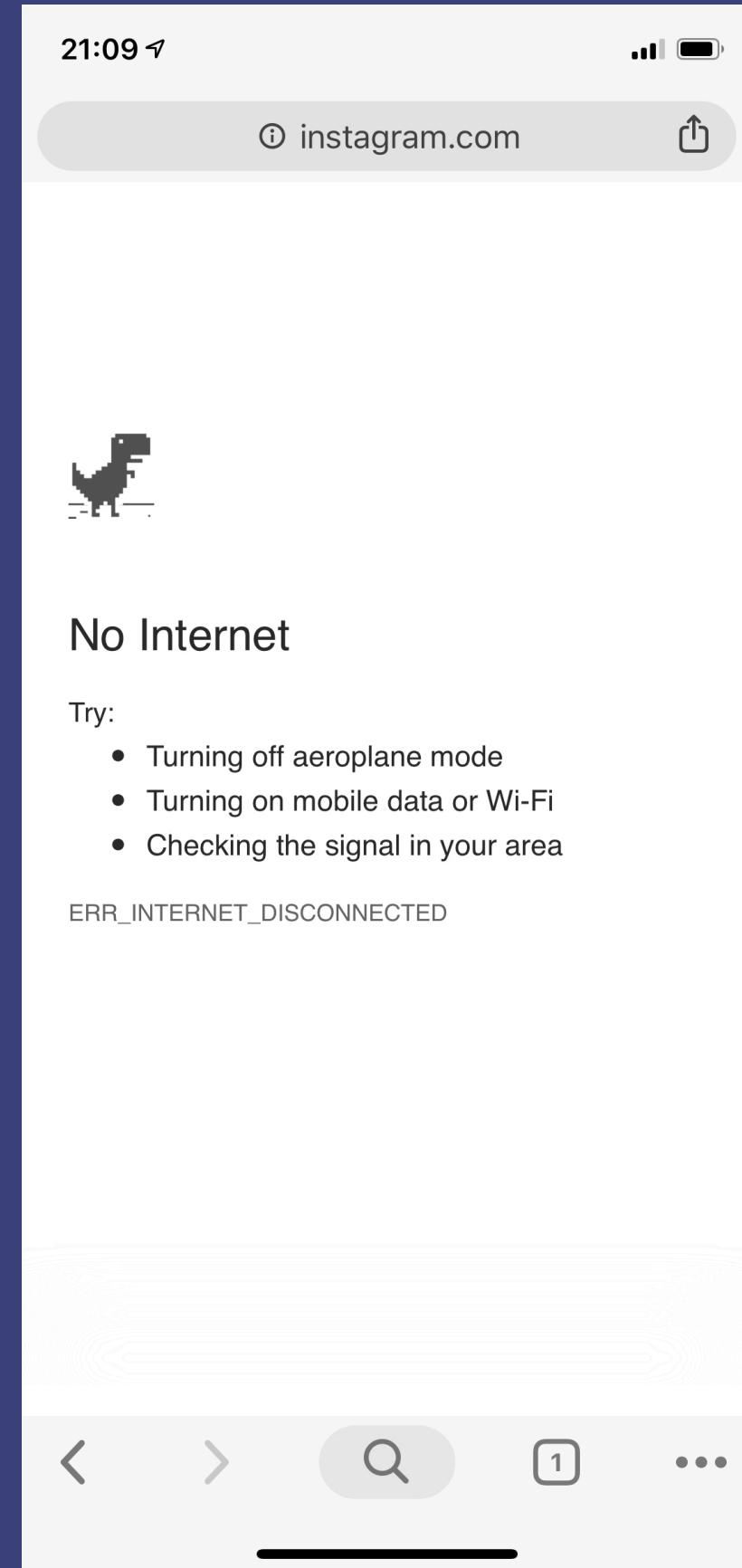
- Functionaliteit komt dichter bij native
- Offline caching, push notificaties, Web Bluetooth, WebVR, Payment API's

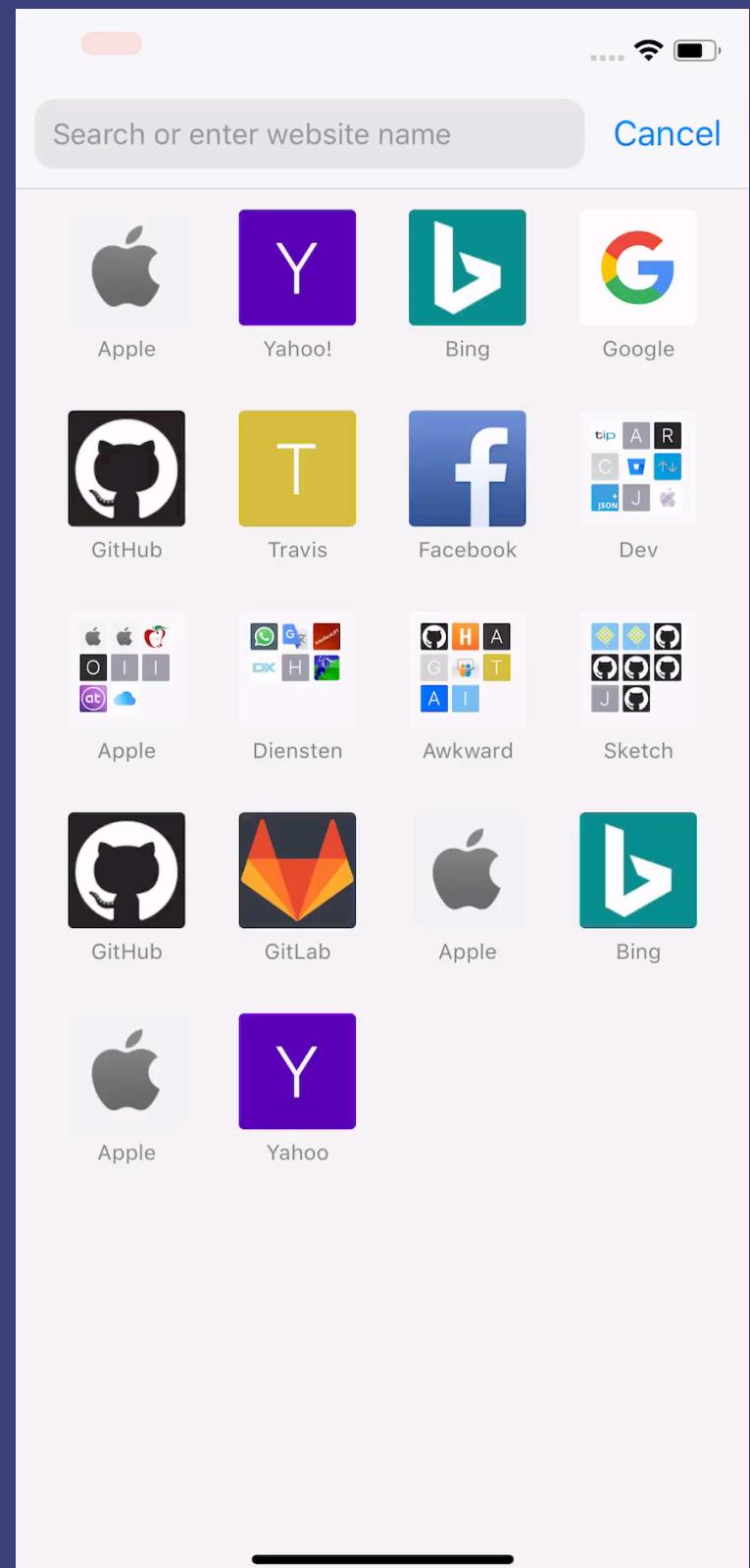
<b>Camera &amp; Microphone</b>	<b>Native Behaviors</b>	<b>Seamless Experience</b>
AUDIO & VIDEO CAPTURE ✓ ADVANCED CAMERA CONTROLS ✓ RECORDING MEDIA ✓ REAL-TIME COMMUNICATION ✓	LOCAL NOTIFICATIONS ✓ PUSH MESSAGES ✓ HOME SCREEN INSTALLATION ✓ FOREGROUND DETECTION ✓ PERMISSIONS ✓	OFFLINE MODE ✓ BACKGROUND SYNC ✓ INTER-APP COMMUNICATION ✘ PAYMENTS ✓ CREDENTIALS ✓
<b>Surroundings</b>	<b>Operating System</b>	<b>Location &amp; Position</b>
BLUETOOTH ✓ USB ✓ NFC ✘ AMBIENT LIGHT ✘	OFFLINE STORAGE ✓ FILE ACCESS ✓ CONTACTS ✘ SMS ✘ STORAGE QUOTAS ✓ TASK SCHEDULING ✘	GEOLOCATION ✓ GEOFENCING ✘ DEVICE POSITION ✓ DEVICE MOTION ✓ PROXIMITY SENSORS ✘
<b>Device Features</b>	<b>Input</b>	<b>Screen &amp; Output</b>
NETWORK TYPE & SPEED ✓ ONLINE STATE ✓ VIBRATION ✓ BATTERY STATUS ✓ DEVICE MEMORY ✓	TOUCH GESTURES ✓ SPEECH RECOGNITION ✓ CLIPBOARD (COPY & PASTE) ✓ POINTING DEVICE ADAPTATION ✓	VIRTUAL & AUGMENTED REALITY ✘ FULLSCREEN ✓ SCREEN ORIENTATION & LOCK ✓ WAKE LOCK ✘ PRESENTATION FEATURES ✓

<https://whatwebcando.today>

# App-achtige ervaring

- Snel
- Werkt offline
- Op het startscherm
- Push notificaties





# Voordelen van PWA's

- Veilig
- Snel
- Goedkoper om te bouwen
- Geen installatie proces
- Vindbaarheid
- Deelbaarheid
- Onafhankelijkheid

Twitter Lite - Apps op Google Play X +

https://play.google.com/store/apps/details?id=com.twitter.and

Google Play Zoeken

Apps Categorieën Homepage Populairst Nieuwe releases

Mijn apps Winkelen Games Gezin Keuze van de redactie

Account Betaalmethoden Mijn abonnementen Tegoed inwisselen Cadeaukaart kopen Mijn verlanglijstje Mijn Play-activiteit Gids voor ouders

# Twitter Lite

Twitter, Inc. Nieuws en tijdschriften ★★★★★ 26.419

3 PEGI 3

Bevat advertenties

⚠ Je hebt geen apparaten.

Toevoegen aan verlanglijstje

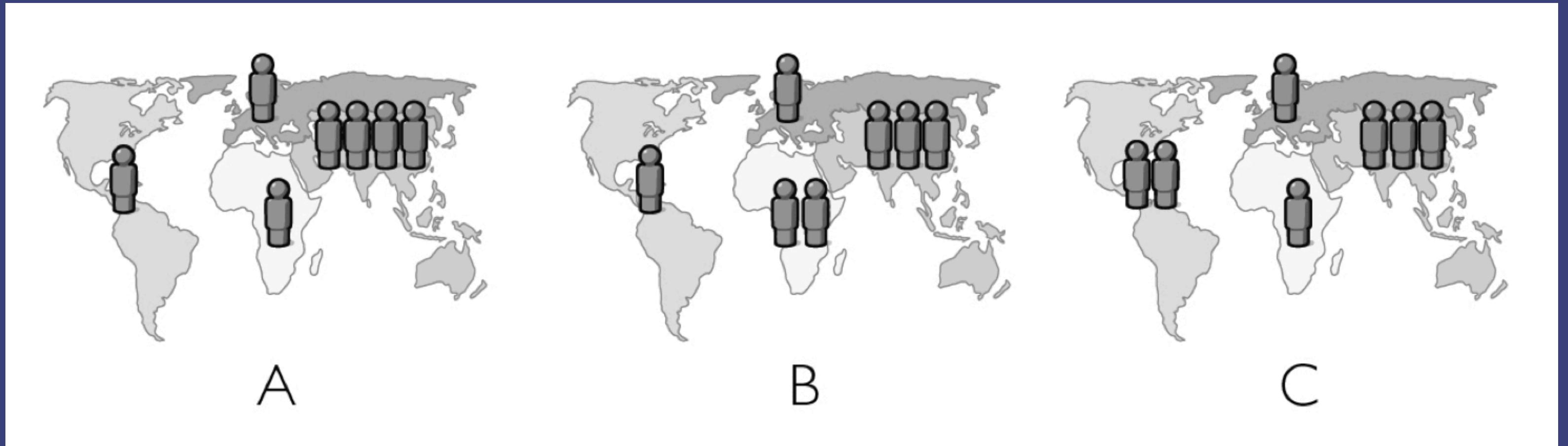
**Data Saver** Use less data

**Timeline** Stay in the know

**Profile** Follow your interests

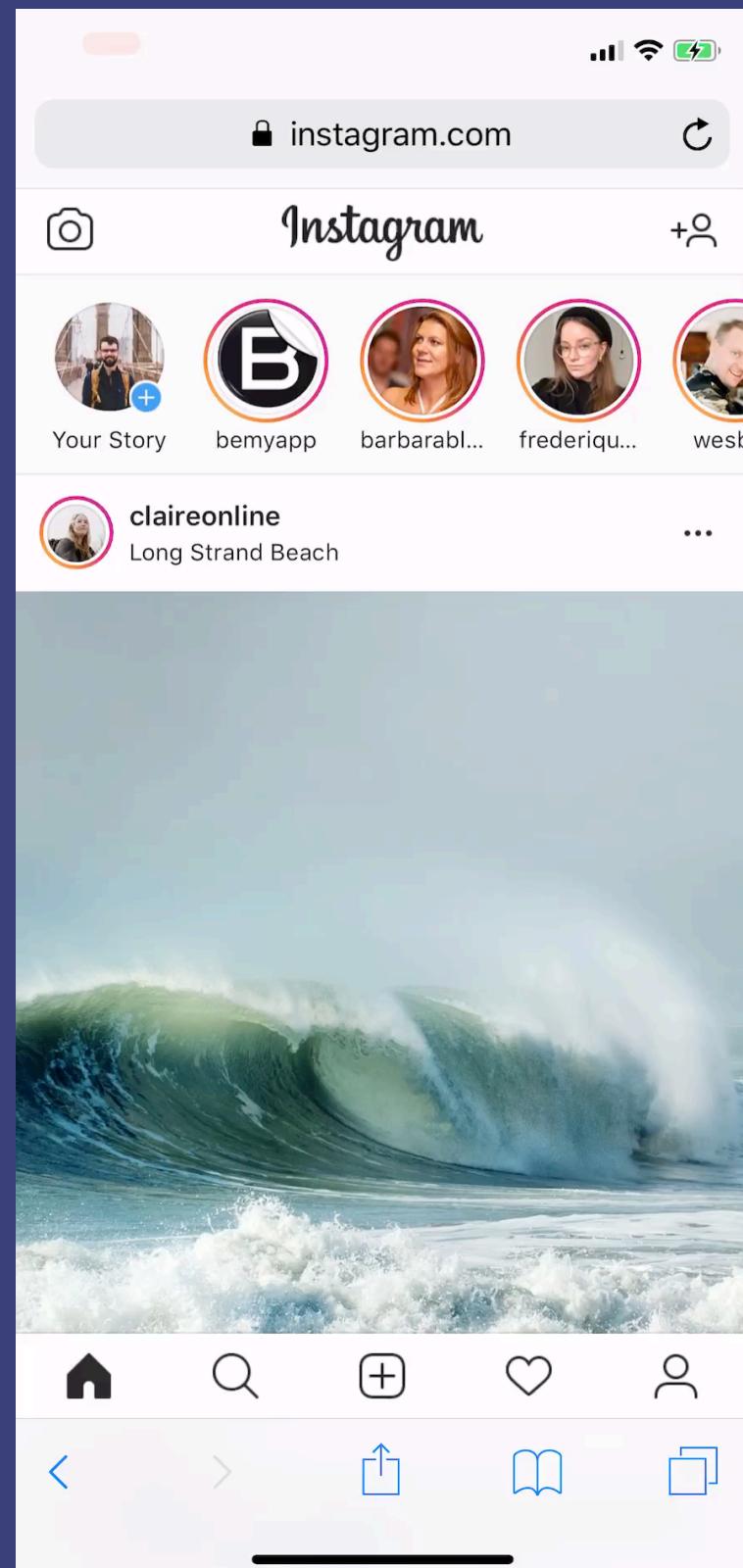
**Explore** See what's new

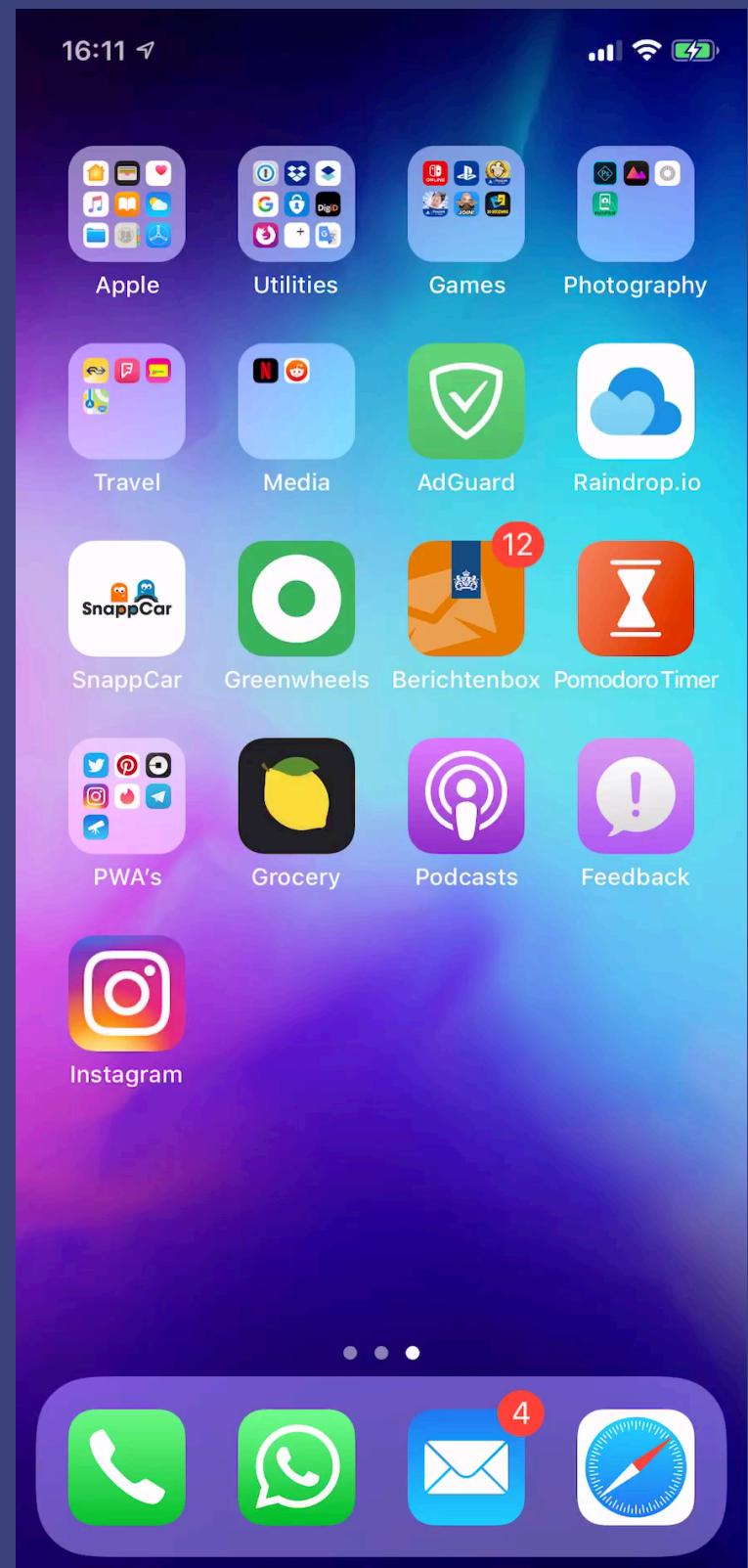
The screenshot shows the Google Play Store page for the Twitter Lite app. The main content area displays four cards illustrating the app's features: 'Data Saver' (using less data), 'Timeline' (staying in the know), 'Profile' (following interests), and 'Explore' (seeing what's new). Each card includes a small preview image of the app's UI. The left sidebar of the Google Play interface is visible, showing navigation links like 'Mijn apps', 'Winkelen', and various account settings. The top navigation bar of the browser window also shows the URL and some standard browser controls.

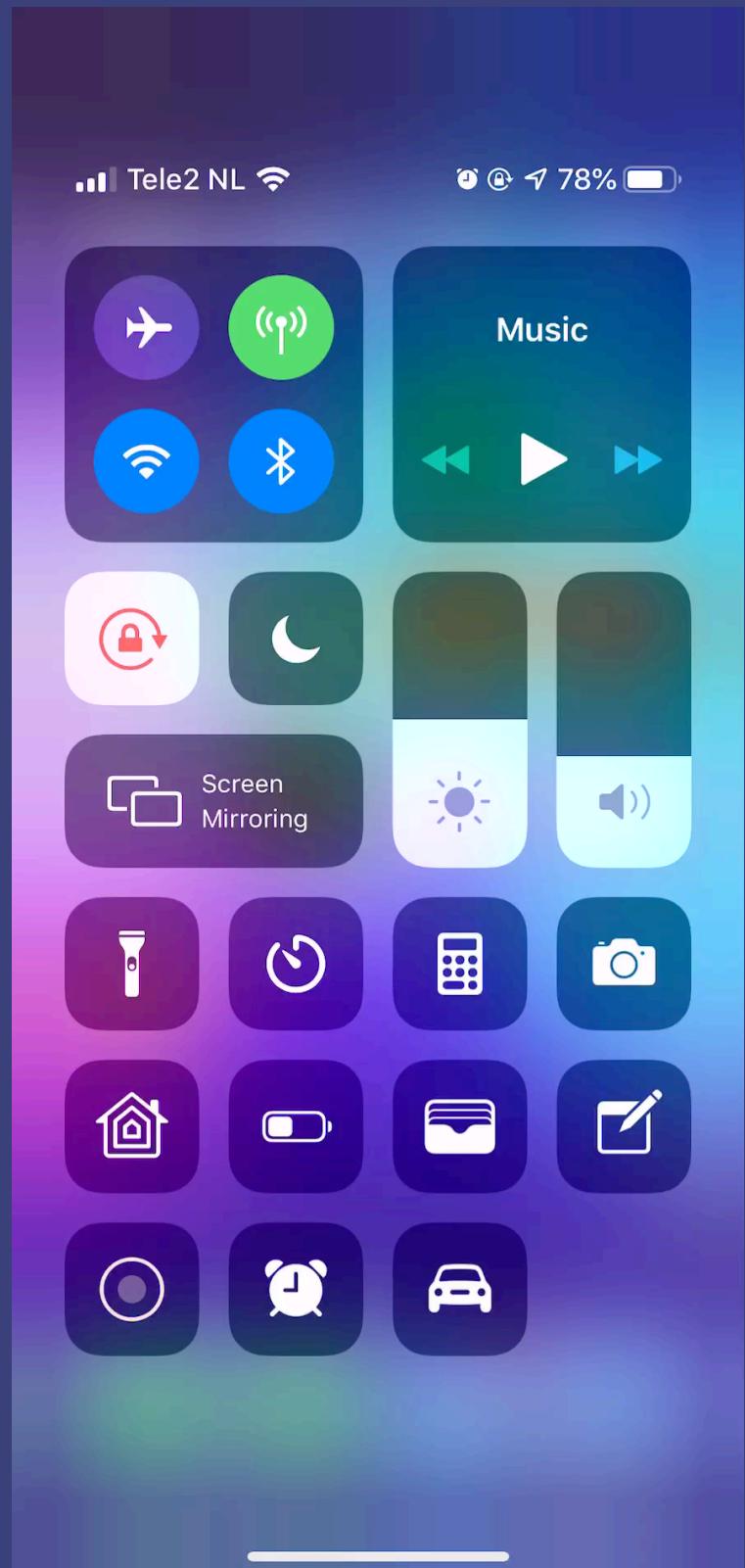


# Voorbeelden van PWA's

- Uber
- Twitter
- Instagram
- Pinterest







Hoe bouw je  
een PWA?

# Eisen aan een PWA

- Het moet *HTTPS* toepassen
- Het moet een *Web App Manifest* bevatten
- Het moet een *service worker* toepassen
- PWA checklist<sup>1</sup>

<sup>1</sup> <https://developers.google.com/web/progressive-web-apps/checklist>

# HTTPS

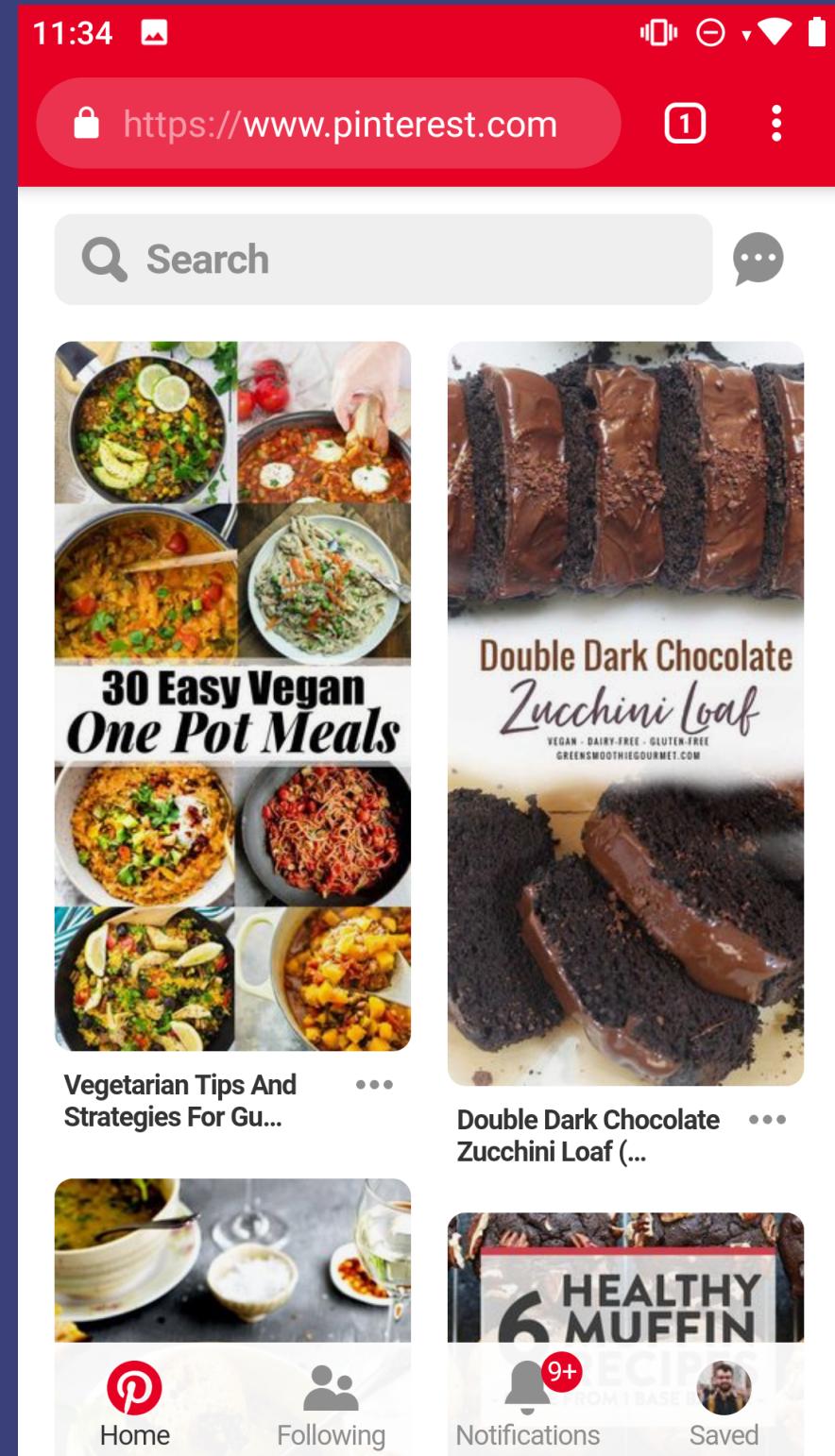
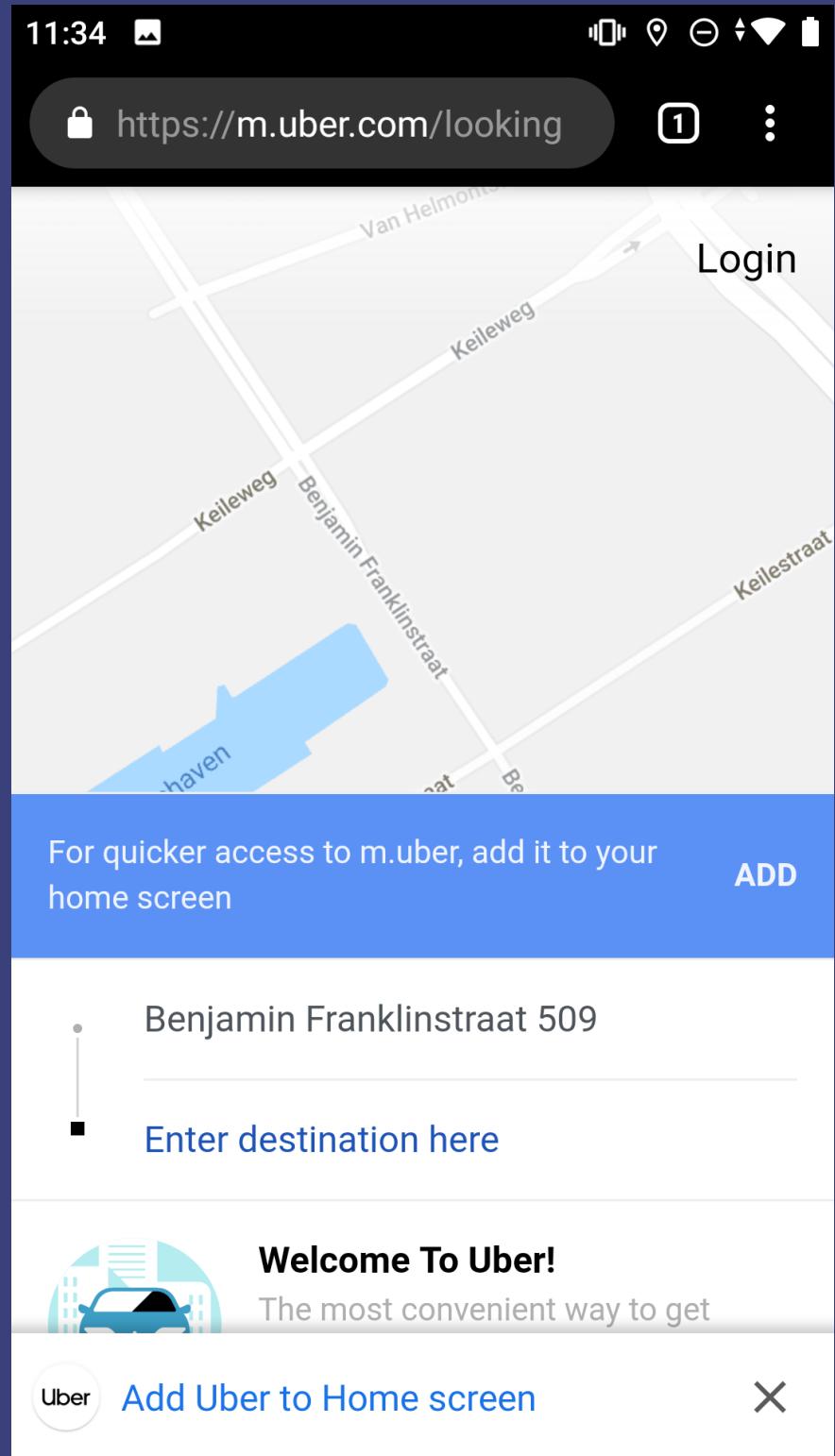
- Bewaakt integriteit van de website
- Vereiste voor moderne web API's
- Netlify, Now.sh, Firebase Hosting

# Web App Manifest

- Thema kleuren
- App icons

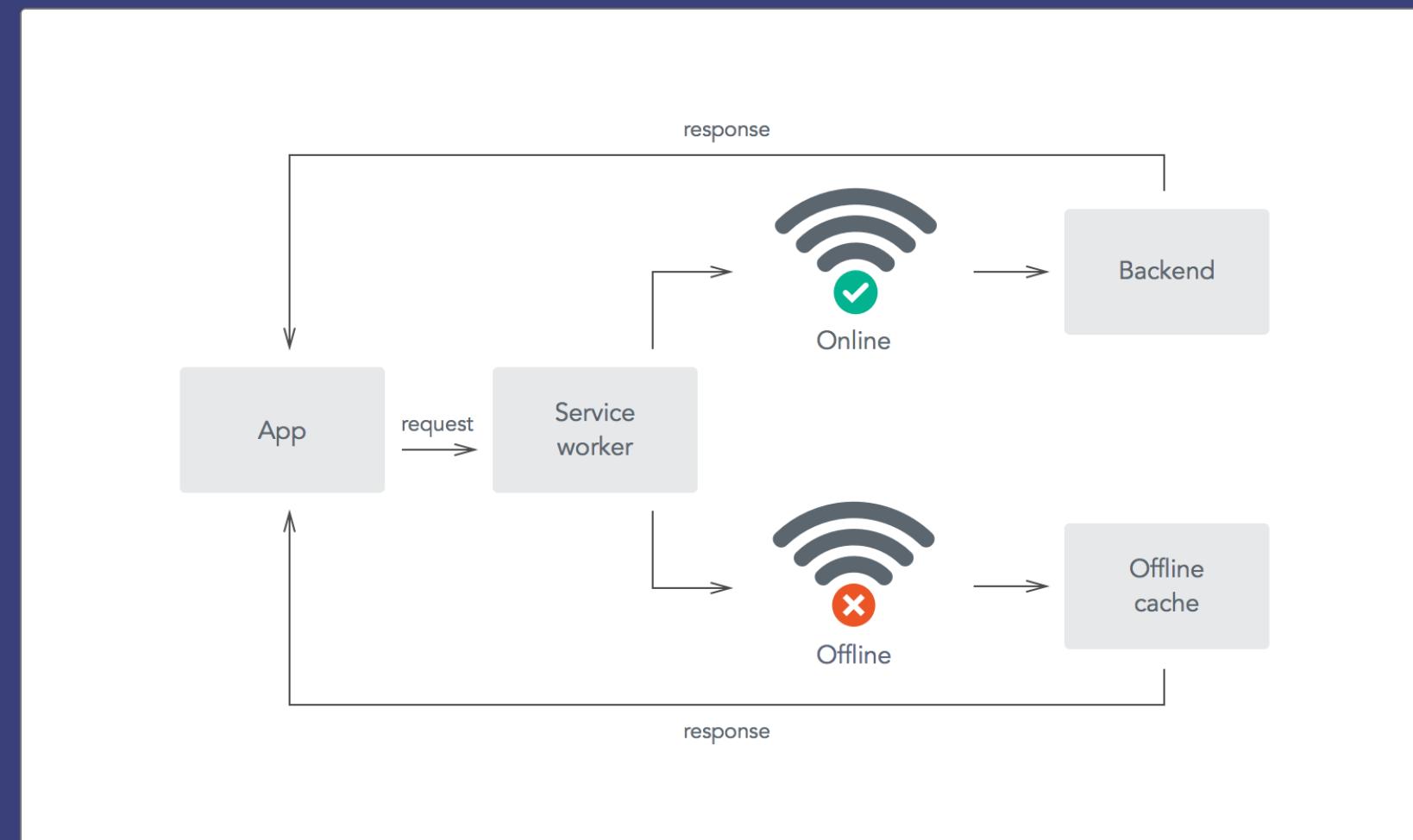
# Web App Manifest

```
{  
  "short_name": "Maps",  
  "name": "Google Maps",  
  "icons": [  
    {  
      "src": "/images/icons-192.png",  
      "type": "image/png",  
      "sizes": "192x192"  
    },  
    ...  
  ],  
  "start_url": "/maps/?source=pwa",  
  "background_color": "#3367D6",  
  "display": "standalone",  
  "scope": "/maps/",  
  "theme_color": "#3367D6"  
}
```



# Service worker

Proxy tussen client en server



# Service worker

- Draait in de achtergrond
- Gebaseerd op Web Workers
- Offline caching, push notificaties, background sync

# Service worker – Cache API

```
caches.open('my-cache').then(cache => {  
  // do something with cache...  
})
```

- cache.match(request, options)
- cache.put(request, response)
- cache.delete(request, options)

# Progressive Web Cats

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Progressive Web Cats</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <link rel="stylesheet" type="text/css" media="screen" href="main.css" />
  </head>
  <body>
    <nav>
      <a href="/">Home</a>
      <a href="/about.html">About</a>
    </nav>
    <script src="main.js"></script>
  </body>
</html>
```

```
const imageContainer = document.querySelector('.image-container')
const image = new Image()
imageContainer.appendChild(image)

fetch('https://cataas.com/cat', { cache: 'no-store' })
  .then(response => response.blob())
  .then(blob => URL.createObjectURL(blob))
  .then(url => {
    image.src = url
  })
```

<https://conservative-web-cats.netlify.com>

# manifest.webmanifest

```
{  
  "name": "Progressive Web Cats",  
  "short_name": "PWC",  
  "description": "View random Progressive Web Cats",  
  "display": "standalone",  
  "background_color": "#fff",  
  "start_url": "/",  
  "icons": [  
    {  
      "src": "images/icons/icon-192x192.png",  
      "sizes": "192x192",  
      "type": "image/png"  
    },  
    {  
      "src": "images/icons/icon-512x512.png",  
      "sizes": "512x512",  
      "type": "image/png"  
    }  
  ]  
}
```

# service-worker.js (pre-caching)

```
const CACHE_NAME = `site-cache-v1`
const urlsToCache = ['/', '/about.html', '/main.css', '/main.js']

self.addEventListener('install', event => {
  console.log('[ServiceWorker] Install')
  event.waitUntil(
    caches.open(CACHE_NAME).then(cache => {
      console.log('[ServiceWorker] Caching app shell')
      return cache.addAll(urlsToCache)
    })
  )
}

self.addEventListener('fetch', event => {
  event.respondWith(
    caches.match(event.request).then(response => {
      return response || fetch(event.request)
    })
  )
})
```



**<https://5c82167bcc526231c3584dd1--progressive-web-cats.netlify.com>**

# service-worker.js (pre-caching + dynamic caching)

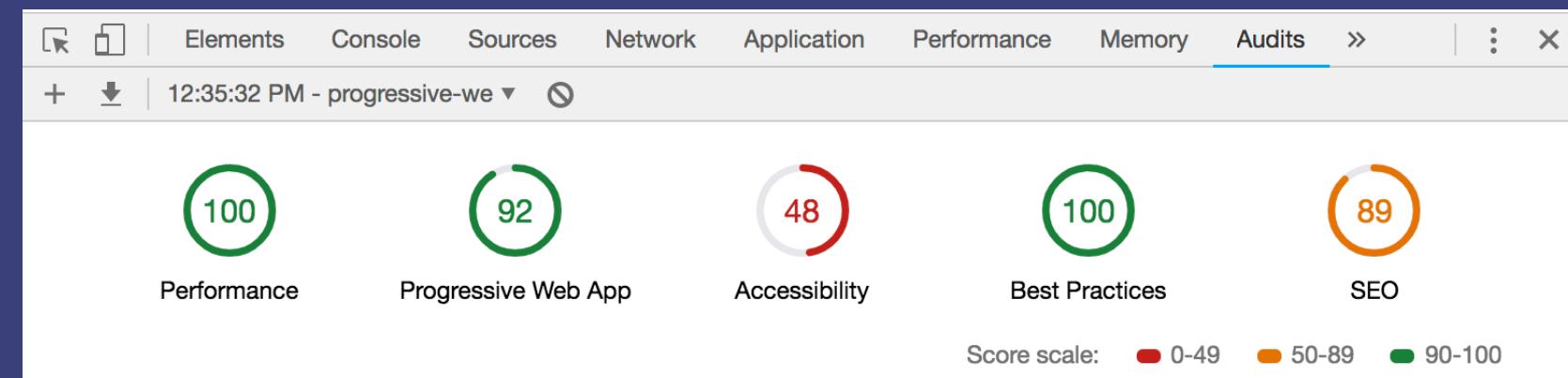
```
self.addEventListener('fetch', event => {
  const requestURL = new URL(event.request.url)

  if (requestURL.hostname === 'cataas.com') {
    event.respondWith(
      fetch(event.request)
        .then(response => {
          if (!response || response.status !== 200) return response

          const responseToCache = response.clone()
          caches.open(CACHE_NAME).then(cache => {
            cache.put(event.request, responseToCache)
          })
          return response
        })
        .catch(() => {
          return caches.match(event.request)
        })
    )
    return
  }

  event.respondWith(
    caches.match(event.request).then(response => {
      return response || fetch(event.request)
    })
  )
})
```

<https://progressive-web-cats.netlify.com>



The screenshot shows the Chrome DevTools Application tab with the Service Workers panel selected. It displays information for the service worker at [progressive-web-cats.netlify.com](https://progressive-web-cats.netlify.com). The service worker is identified as #7974 and is currently running. There are options to test push messages and sync data. A note indicates the service worker was received on 1/1/1970, 1:00:00 AM.

The screenshot shows the Chrome DevTools Performance tab with detailed metrics and diagnostics. Key metrics include First Contentful Paint (0.8 s), Speed Index (0.8 s), Time to Interactive (0.9 s), and Estimated Input Latency (10 ms). The Diagnostics section provides more information about the performance of the application, and the Passed audits section lists 21 successful audits.

## Performance

### Metrics

Metric	Value	Status	Description
First Contentful Paint	0.8 s	✓	First Meaningful Paint
Speed Index	0.8 s	✓	First CPU Idle
Time to Interactive	0.9 s	✓	Estimated Input Latency

**View Trace**

Values are estimated and may vary.

### Diagnostics

More information about the performance of your application.

- 1 Minimize Critical Requests Depth

**Passed audits**

21 audits

## Progressive Web App

These checks validate the aspects of a Progressive Web App, as specified by the baseline [PWA Checklist](#).

- 1 Is not configured for a custom splash screen  
Failures: Manifest does not have `theme\_color`.
- 2 Address bar does not match brand colors

# Nadelen van PWA's

- Slechte ondersteuning op iOS
- Ontbrekende functionaliteiten
- Verdienmodel

***There are only two hard things in Computer Science:  
cache invalidation and naming things***

– Phil Karlton

# Frameworks

- React (create-react-app)<sup>2</sup>
- Preact (preact-cli)<sup>3</sup>
- Vue CLI PWA plugin<sup>4</sup>

<sup>2</sup> <https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app>

<sup>3</sup> <https://preactjs.com>

<sup>4</sup> <https://www.npmjs.com/package/@vue/cli-plugin-pwa>

# Resources

- PWA checklist<sup>5</sup>
- Offline cookbook<sup>6</sup>
- HTTP203 show op YouTube<sup>7</sup>
- Appscoope<sup>8</sup>

<sup>5</sup> <https://developers.google.com/web/progressive-web-apps/checklist>

<sup>6</sup> <https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook>

<sup>7</sup> <https://www.youtube.com/user/ChromeDevelopers>

<sup>8</sup> <https://www.appsco.pe>



Twitter: @timonvspronson  
GitHub: [github.com/timonvs](https://github.com/timonvs)