Progressive Web Apps

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

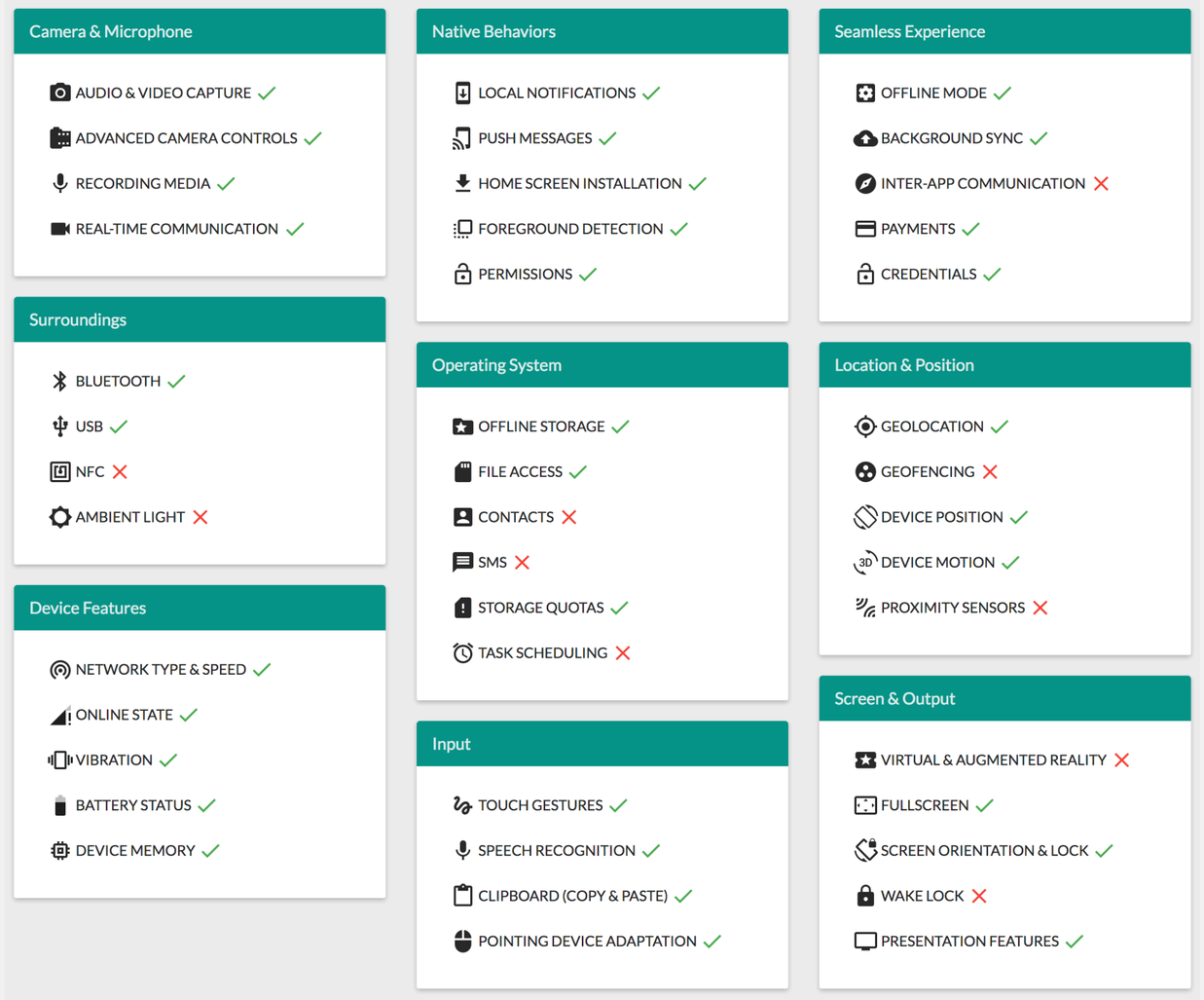
→ Extra functionaliteit voor modernere browsers

→ Polyfills

Moderne web API's

→ Functionaliteit komt dichter bij native

→ Offline caching, push notificaties, Web Bluetooth, WebVR, Payment API's



https://whatwebcando.today

App-achtige ervaring

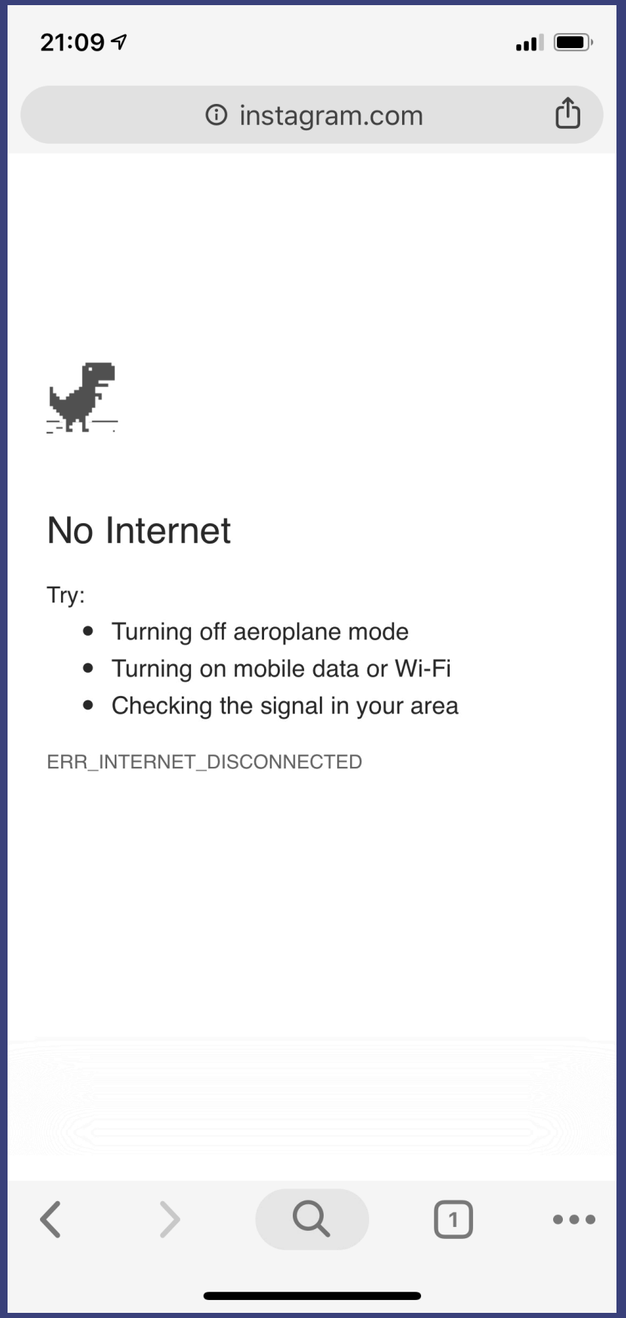
→ Snel

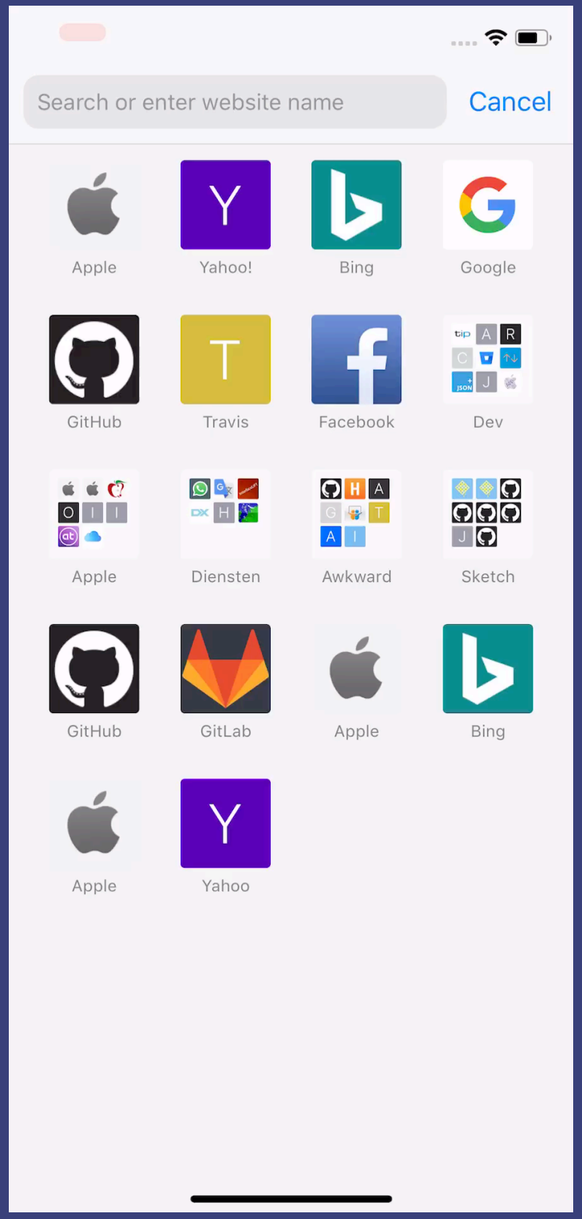
→ Werkt offline

→ Op het startscherm

→ Push notificaties


21:07
Instagram
No Internet Connection
Your Story
anoukmelis_
bemyapp
barbarablee...
crowded_kitchen


21:09
instagram.com
No Internet
Try:
Turning off aeroplane mode
Turning on mobile data or Wi-Fi
Checking the signal in your area
ERR_INTERNET_DISCONNECTED


Search or enter website name
Cancel
Apple
Yahoo!
Bing
Google
GitHub
Travis
Facebook
Dev
Apple
Diensten
Awkward
Sketch
GitHub
GitLab
Apple
Bing
Apple
Yahoo

Voordelen van PWA's

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

Twitter Lite - Apps op Google P

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

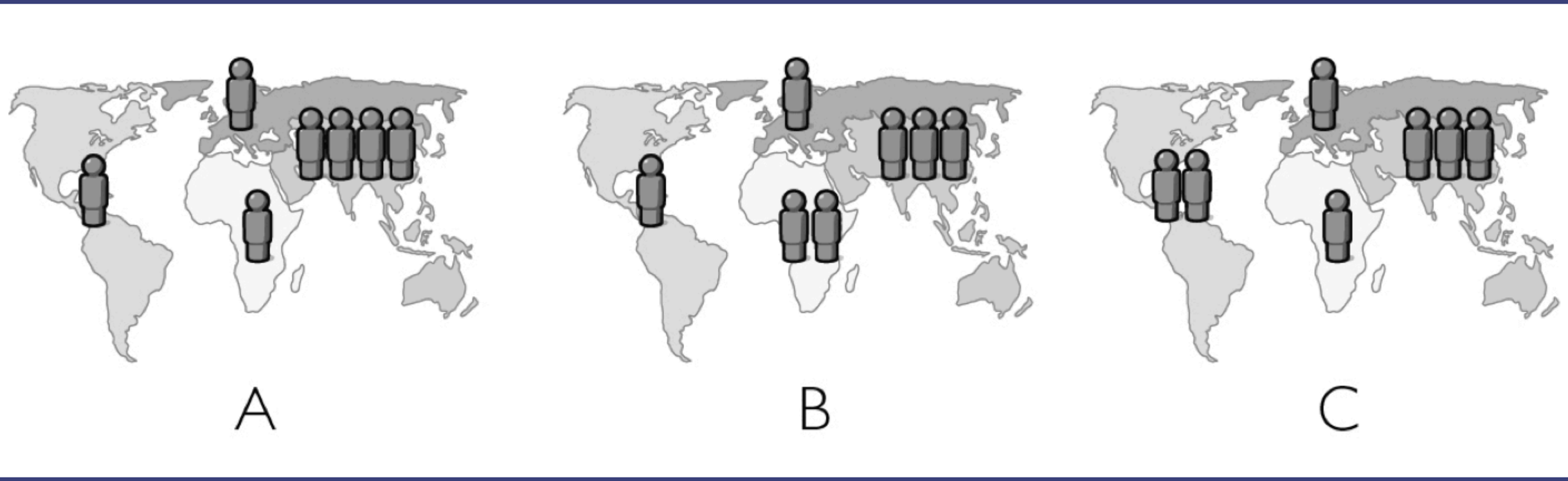
★★★★☆ 26.419

PEGI 3

Bevat advertenties

Je hebt geen apparaten.

Toevoegen aan verlanglijstje



Bron: gapminder.org

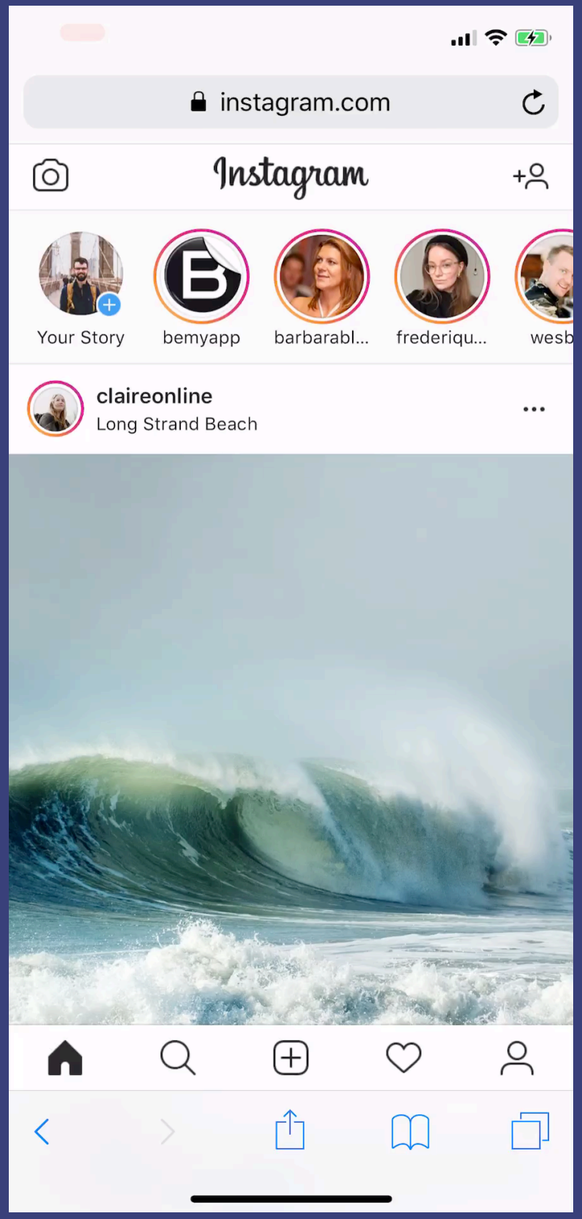
Voorbeelden van PWA's

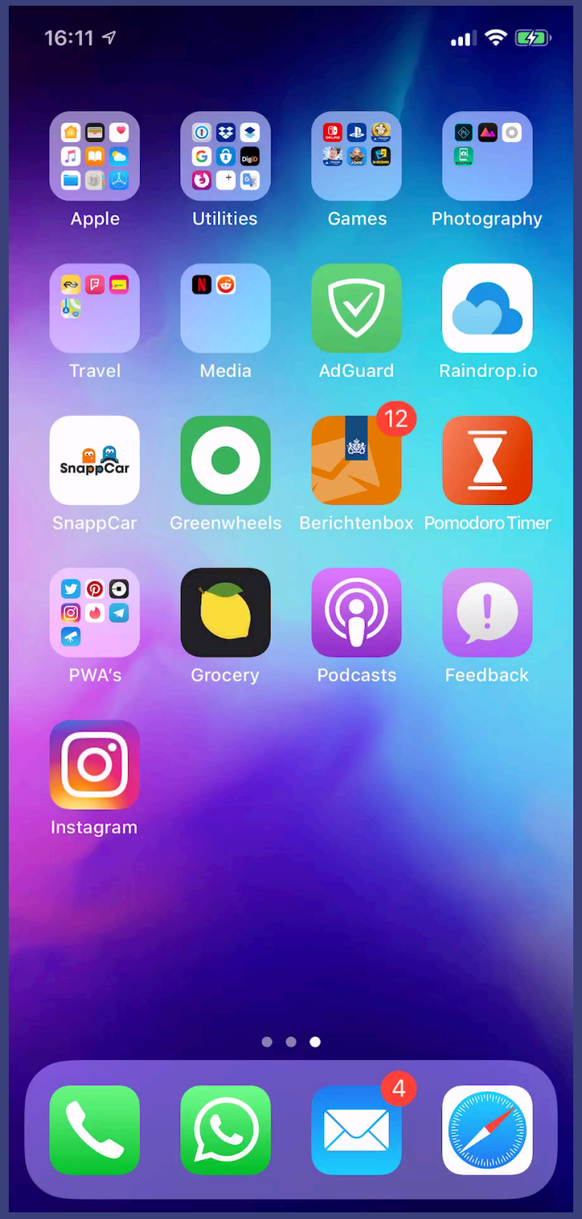
→ Uber

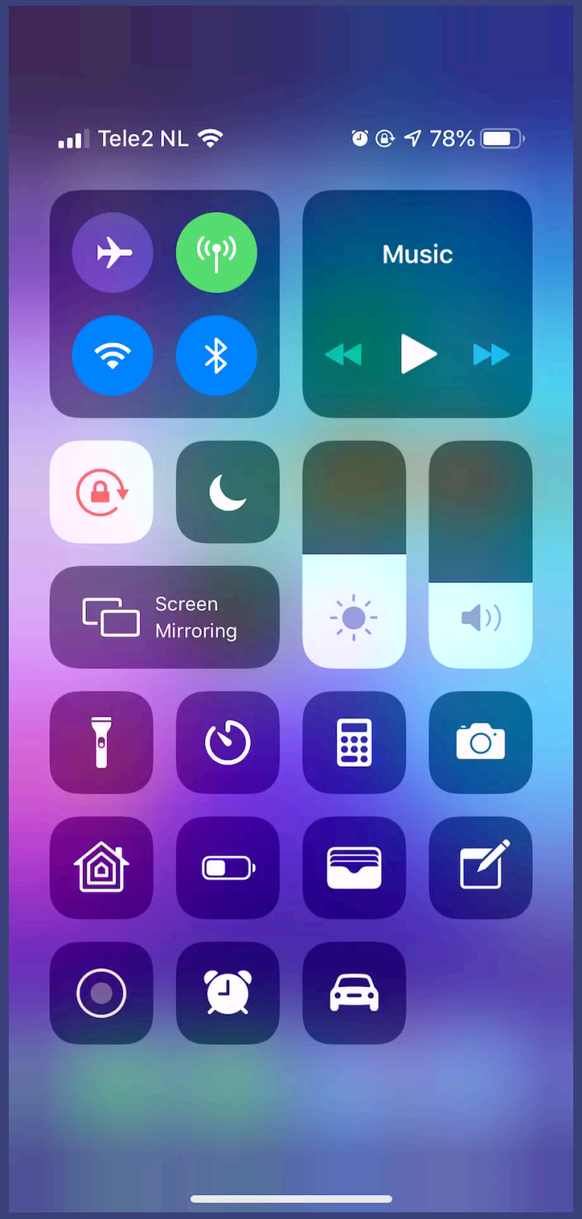
→ Twitter

→ Instagram

→ Pinterest


instagram.com
Instagram
Your Story
bemyapp
barbarabl...
frederiqu...
wesb
claireonline
Long Strand Beach


16:11
Apple
Utilities
Games
Photography
Travel
Media
AdGuard
Raindrop.io
SnappCar
SnappCar
Greenwheels
12
Berichtenbox
PomodoroTimer
PWA's
Grocery
Podcasts
Feedback
Instagram
4


Tele2 NL
78%
Music
Screen
Mirroring

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[1]

[1] 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

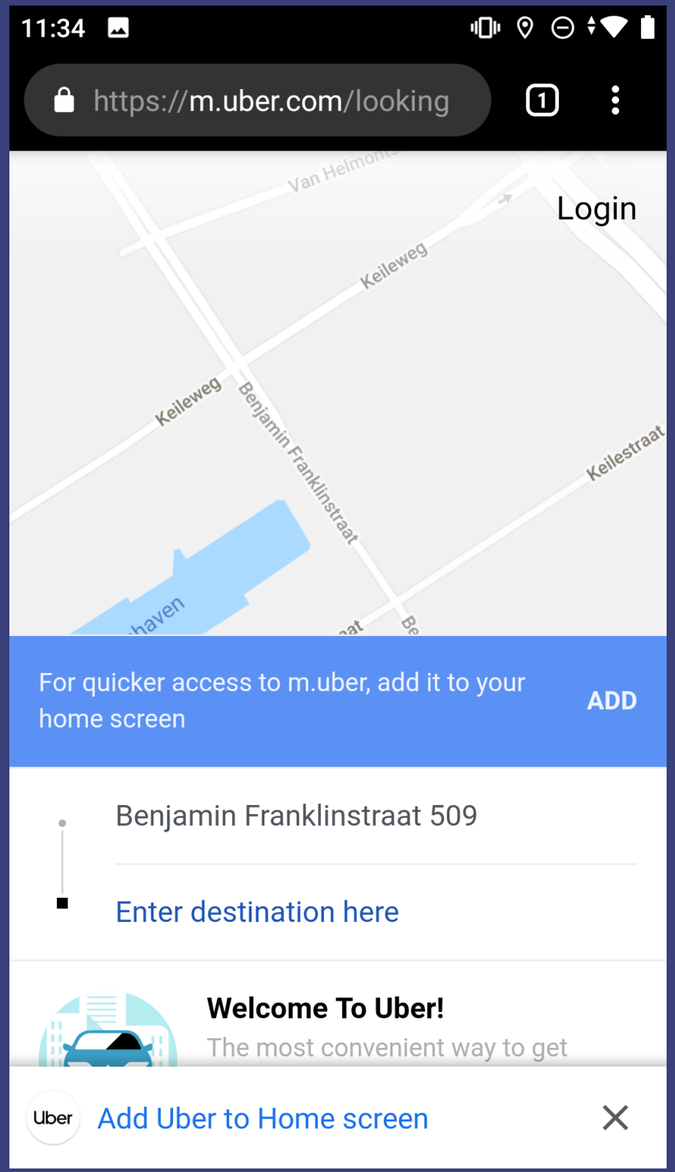
Web App Manifest

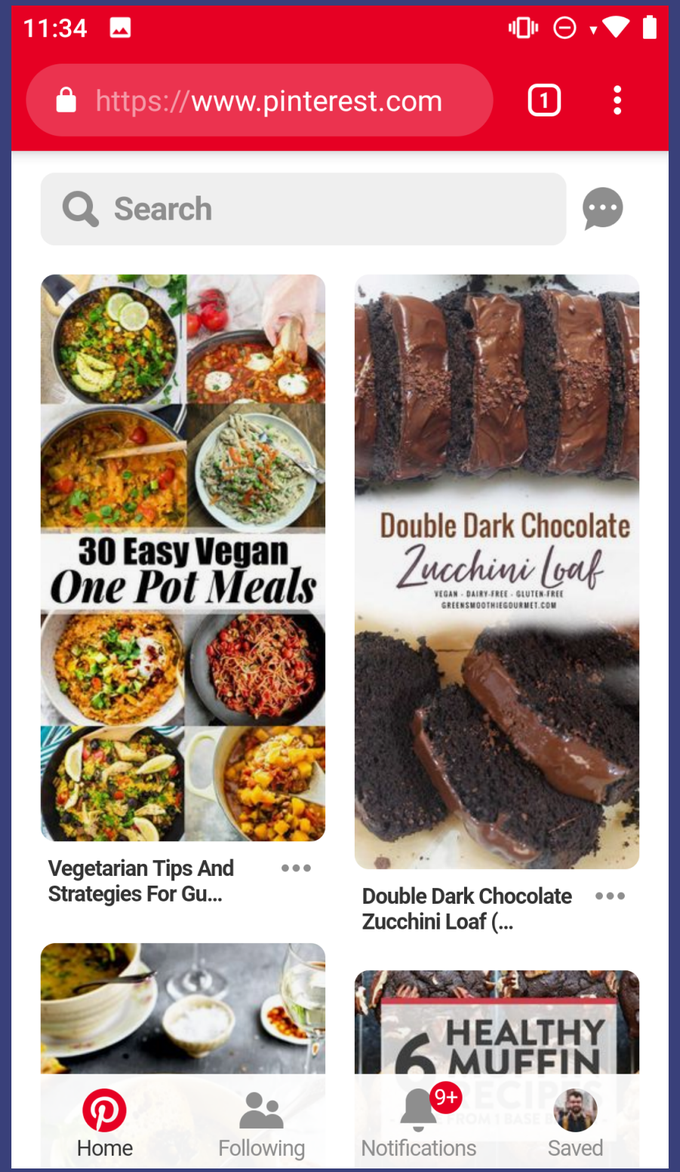
→ Thema kleuren

→ App icons

Web App Manifest

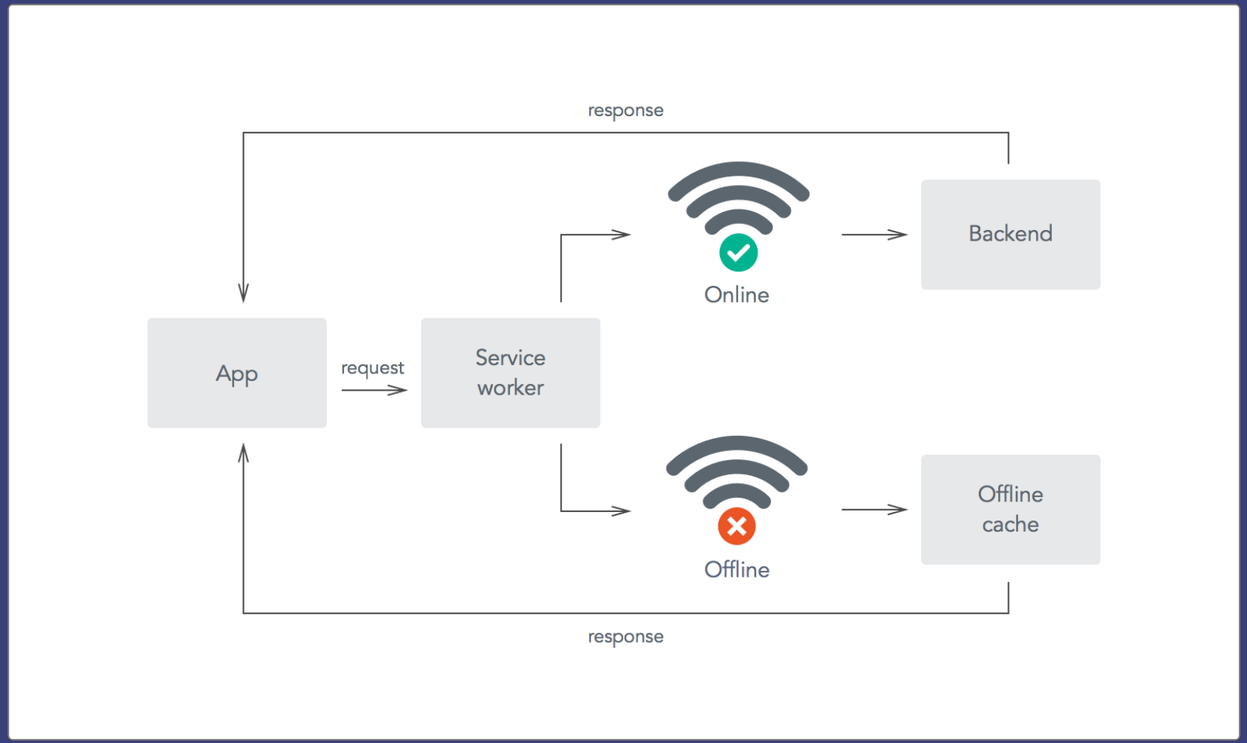
```
{
  "short_name": "Maps",
  "name": "Google Maps",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    ...
  ],
  "start_url": "/maps/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/maps/",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


11:34
https://m.uber.com/looking
1
Login
Keileweg
Keileweg
Benjamin Franklinstraat
Keilestraat
For quicker access to m.uber, add it to your home screen
ADD
Benjamin Franklinstraat 509
Enter destination here
Welcome To Uber!
The most convenient way to get
Uber
Add Uber to Home screen


11:34
https://www.pinterest.com
1
Search
30 Easy Vegan One Pot Meals
Vegetarian Tips And Strategies For Gu...
Double Dark Chocolate Zucchini Loaf
Double Dark Chocolate Zucchini Loaf (...
HEALTHY MUFFIN
Home
Following
9+
Notifications
Saved

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)
    })
  )
})
```

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)
    })
  )
})
```

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)
    })
  )
})
```

```html
<!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" />
    <link rel="manifest" href="/manifest.webmanifest" />
    <link rel="apple-touch-icon" sizes="180x180" href="images/icons/apple-touch-icon-180x180.png" />
  </head>
  <body>
    <nav>
      <a href="/">Home</a>
      <a href="/about.html">About</a>
    </nav>
    <div class="image-container"></div>
    <script>
      navigator.serviceWorker
        .register('service-worker.js')
        .then(() => {
          console.log('succesfully registered service worker!')
        })
        .catch(() => {
          console.log('failed to register service worker')
        })
    </script>
    <script src="main.js"></script>
  </body>
</html>
```

```
<!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" />
    <link rel="manifest" href="/manifest.webmanifest" />
    <link rel="apple-touch-icon" sizes="180x180" href="images/icons/apple-touch-icon-180x180.png" />
  </head>
  <body>
    <nav>
      <a href="/">Home</a>
      <a href="/about.html">About</a>
    </nav>
    <div class="image-container"></div>
    <script>
      navigator.serviceWorker
        .register('service-worker.js')
        .then(() => {
          console.log('succesfully registered service worker!')
        })
        .catch(() => {
          console.log('failed to register service worker')
        })
    </script>
    <script src="main.js"></script>
  </body>
</html>
```

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)
    })
  )
})
```

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)
    })
  )
})
```

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)
    })
  )
})
```

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)
    })
  )
})
```

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)
    })
  )
})
```

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)
    })
  )
})
```

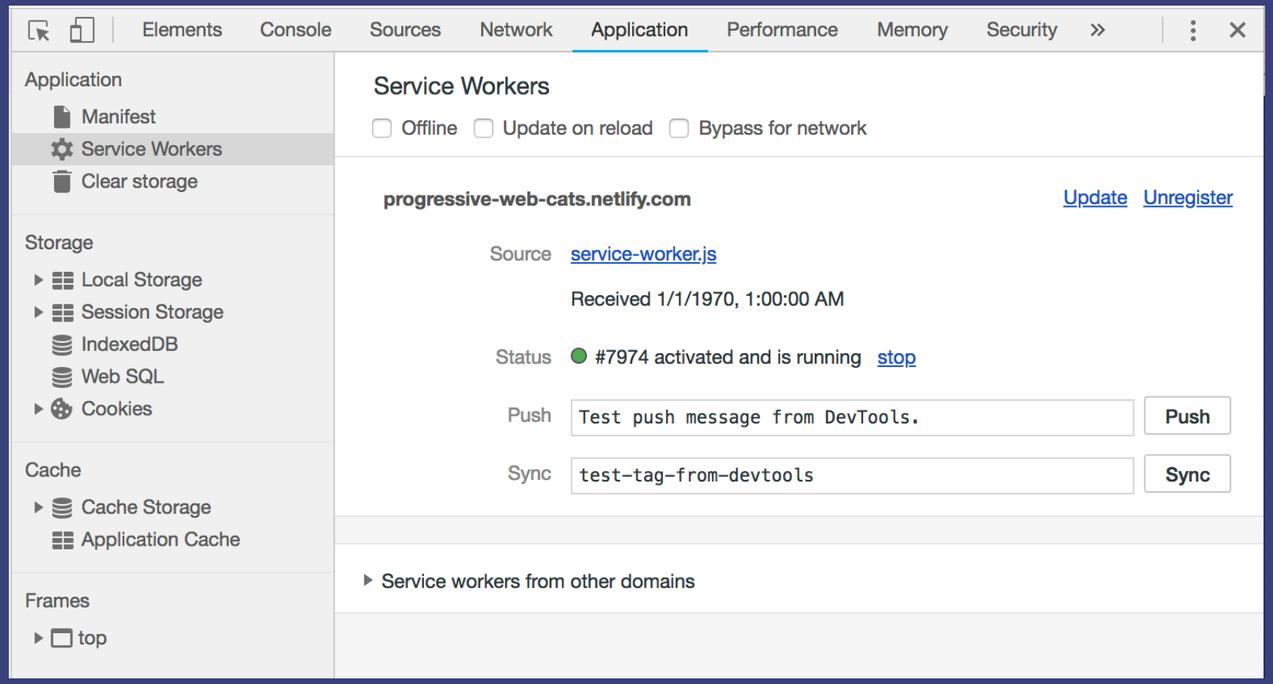
service-worker.js (pre-caching + dynamic caching)

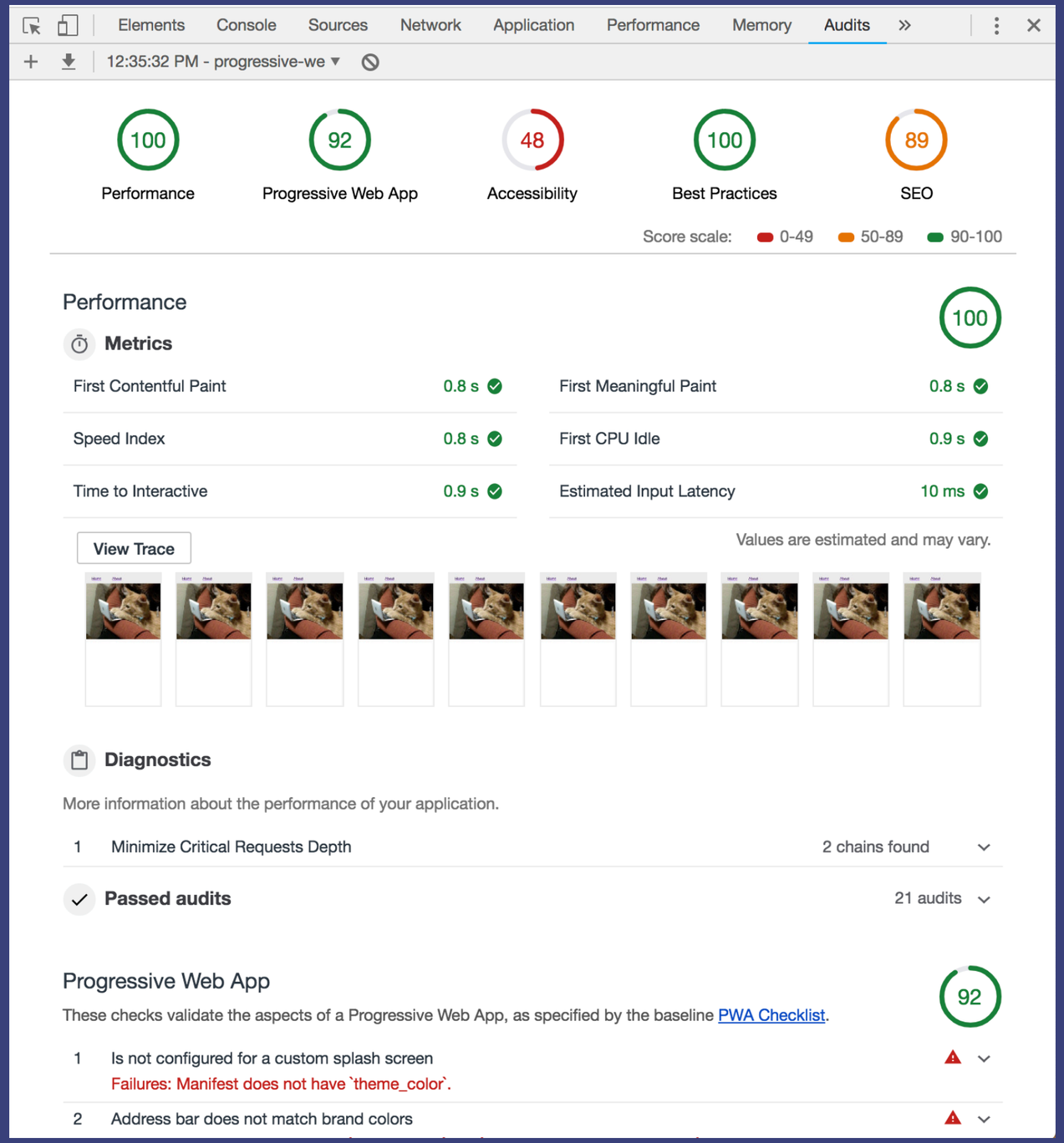
```
self.addEventListener('fetch', event => {
  const requestURL = new URL(event.request.url)

  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)
    })
  )
})
```

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


Elements
Console
Sources
Network
Application
Performance
Memory
Security
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
Cache
Cache Storage
Application Cache
Frames
top
Service Workers
Offline
Update on reload
Bypass for network
progressive-web-cats.netlify.com
Update
Unregister
Source
service-worker.js
Received 1/1/1970, 1:00:00 AM
Status
#7974 activated and is running
stop
Push
Test push message from DevTools.
Push
Sync
test-tag-from-devtools
Sync
Service workers from other domains


Elements
Console
Sources
Network
Application
Performance
Memory
Audits
12:35:32 PM - progressive-we
100
Performance
92
Progressive Web App
48
Accessibility
100
Best Practices
89
SEO
Score scale:
0-49
50-89
90-100
Performance
100
Metrics
First Contentful Paint
0.8 s
First Meaningful Paint
0.8 s
Speed Index
0.8 s
First CPU Idle
0.9 s
Time to Interactive
0.9 s
Estimated Input Latency
10 ms
View Trace
Values are estimated and may vary.
Diagnostics
More information about the performance of your application.
1
Minimize Critical Requests Depth
2 chains found
Passed audits
21 audits
Progressive Web App
92
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)[2]

→ Preact (preact-cli)[3]

→ Vue CLI PWA plugin[4]

[2] https://facebook.github.io/create-react-app/docs/making-a-progressive-web-app

[3] https://preactjs.com

[4] https://www.npmjs.com/package/@vue/cli-plugin-pwa

Resources

→ PWA checklist[5]

→ Offline cookbook[6]

→ HTTP203 show op YouTube[7]

→ Appscope[8]

[5] https://developers.google.com/web/progressive-web-apps/checklist

[6] https://developers.google.com/web/fundamentals/instant-and-offline/offline-cookbook

[7] https://www.youtube.com/user/ChromeDevelopers

[8] https://www.appsco.pe

Twitter: @timonvspronsen
GitHub: github.com/timonvs