

PWA

Progressive Web App

David Schol / 2022

Wat is een PWA?



Progressive Web Application (PWA) is een type web app die zowel als webpagina en als mobiele app op elk apparaat kan werken. Met behulp van standaardtechnologieën is een PWA gericht op het leveren van een native-achtige gebruikerservaring, met een snellere conversie en beter browsen, zelfs bij een slechte internetverbinding.

PWA's zijn geschreven in HTML, CSS en JavaScript. Ze zien eruit als gewone webpagina's en gedragen zich als zodanig. Ze leveren echter ook functionaliteiten die identiek zijn aan die van mobiele apps: ze zijn snel, kunnen offline werken, push-notificaties versturen (buggy) en de functies van gebruikersapparaten gebruiken.

Een PWA is dus altijd een responsieve web applicatie, die er op elke schermgrootte optimaal (mogelijk) uitziet. We hanteren bij de ontwikkeling van een PWA bij voorkeur een 'mobile first' aanpak, zodat iedere weergave zo optimaal mogelijk ontwikkelt wordt.



Voordelen van een PWA



De belangrijkste voordelen van het implementeren van PWA zijn verbeterde snelheid en prestaties, app-achtige UX en multi-platform gebruik. Dit zijn echter maar een paar voorbeelden uit een grote lijst met voordelen. Laten we ze eens allemaal van dichterbij bekijken.

1. Eén app voor alle devices. Maar één keer de ontwikkeltijd, dus sneller op de markt
2. Lagere ontwikkel- en onderhoudskosten
3. Je hoeft geen gebruik te maken van een app store en de afhankelijkheid daarvan
4. Altijd up-to-date, want geen vertraging door app stores
5. Installatie gebeurt door de browser => add to homescreen
6. Makkelijk vindbaar door Google. Voordelen van SEO, maar let op bij client side rendering
7. Push berichten

Vereisten voor een PWA



Om een Progressive Web App op te zetten moet je een aantal zaken regelen:

1. Een beveiligde verbinding (HTTPS)

Voor de PWA moet je zorgen voor een beveiligde server met een HTTPS verbinding. Dit is de manier om zowel gebruikersgegevens te beschermen als een extra beveiligingslaag in te bouwen binnen je website. (Bij het ontwikkelen van de PWA, is localhost ook toegestaan).

2. Service Worker

Dit is een van de belangrijkste elementen van de PWA technologie, die helpt om te bepalen hoe netwerkverzoeken moeten worden afgehandeld. Een service worker is een client-side JavaScript bestand toegevoegd aan je codebase, die op de achtergrond werkt om app-achtige functies na te bootsen.

3. Manifest bestand

Je moet een eenvoudig JSON bestand genereren (PWA manifest bestand) dat de browser voorziet van informatie over het uiterlijk en het gedrag van je PWA, zodra deze is geïnstalleerd op het mobiele apparaat van een gebruiker.

Het bevat basisinformatie zoals de app naam, gebruikte iconen of kleuren, en heeft als doel een native app-achtige ervaring te verbeteren.

HTTPS verbinding

Zorg dat de web app op een server komt te staan die beschikt over de secure omgeving. Je kan hiervoor (bij Fontys) uit twee opties kiezen. Een Linux hosting (hera) of een Windows hosting (luna).

Als je de site op de hera-server zet, dan moet je kiezen voor de private map. Alleen dan staat je site in de secure omgeving. Kies je voor de luna-server, dan maak je altijd gebruik van een secure omgeving. Beide server zijn te bereiken via de selfservice portal.

We gebruiken FileZilla om de bestanden middels het FTP protocol van je computer naar de Fontys server te versturen.

Protocol: FTP - File Transfer Protocol

Host: luna.fhict.nl Port:

Encryption: Require explicit FTP over TLS

Logon Type: Normal

User: i889354

Password:

Transfer mode:

☐ Default ☐ Active ☒ Passive

☐ Limit number of simultaneous connections

Maximum number of connections: 1



 i889354.luna.fhict.nl/pwa/



Local site

Filename ^	Filesize	Filetype	Last modified
..			
icons		Directory	09/13/22 12:31:54
img		Directory	09/13/22 21:09:17
favicon.ico	15406	ico-file	09/13/22 21:40:48
index.html	2349	html-file	09/13/22 23:13:02
manifest.json	811	After Effects JSON	09/13/22 21:18:57
styles.css	1354	CSS style sheet	09/14/22 00:13:01
sw.js	2371	js-file	09/14/22 00:15:31

Remote site

Filename ^	Filesize	Filetype	Last modified
..			
icons		Directory	09/13/22 22:42:00
img		Directory	09/13/22 22:42:00
favicon.ico	15406	ico-file	09/14/22 00:13:00
index.html	2412	html-file	09/14/22 00:13:00
manifest.json	811	After Effects JSON	09/14/22 00:13:00
styles.css	1450	CSS style sheet	09/14/22 00:13:00
sw.js	2436	js-file	09/14/22 00:15:00

Vorbereiden index.html



Als we van een website een web app willen maken, dan dienen we voor een aantal zaken te zorgen.

HTML moet het volgende bevatten:

- viewport
- manifest
- apple-touch-icon
- theme-color
- registratie van de serviceWorker (JavaScript)

CSS

- responsive design (media queries)

Handige bronnen:

- logo.com
- [pexels](https://pexels.com) (free stock photos)
- [photopea](https://photopea.com) (online photo editor)
- [The Net Ninja css grid layouts repo](https://github.com/NetNinja/css-grid-layouts-repo) (multi column)
- [Media S3 repo - starter files](#)

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="styles.css">
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
  <title>David Travels - Finished</title>

  <!-- Files to create a web app -->
  <link rel="manifest" href="manifest.json">
  <link rel="apple-touch-icon" href="icons/apple-touch-icon.png">
  <meta name="theme-color" content="#ffffff">
</head>
```

```
<script>
  // Register the service worker
  if ('serviceWorker' in navigator)
  {
    navigator.serviceWorker.register('sw.js');
  };
</script>
```

Het manifest



Het web app manifest is een JSON bestand (manifest.json) dat definieert hoe de PWA moet worden gezien als een geïnstalleerde applicatie, inclusief de look and feel en het basisgedrag binnen het besturingssysteem.

Het web app manifest is een bestand dat je maakt en de browser vertelt hoe je wilt dat de web applicatie wordt weergegeven als een app in het besturingssysteem.

Een web app manifest is een vereiste om de app te installeren in de browser. Je PWA zal niet installeren zonder dit bestand.

Het manifest kan basisinformatie bevatten, zoals de naam, de pictogram (icon) en de themakleur van de app; uitgebreide voorkeuren, zoals de gewenste oriëntatie en app-snelkoppelingen; en metagegevens uit de catalogus, zoals schermafbeeldingen.

Handige bronnen:

- [Web app manifests](#)
- [Web app manifest generator](#)
- [favicon.io](#)

```
{
  "name": "Weather: Do I need an umbrella?",
  "short_name": "Weather",
  "description": "Weather forecast information",
  "start_url": "/index.html",
  "background_color": "#3367D6",
  "theme_color": "#3367D6",
  "display": "standalone",
  "orientation": "portrait",
  "icons":
  [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ]
}
```

De service worker

Zorg dat het sw.js bestand in de root van je web app zit. De service worker werkt vanaf de plaats waar deze staat.

Wat doet een service worker:

- Precached de HTML, JavaScript en CSS bestanden die nodig zijn om deze pagina offline weer te geven. (Probeer het uit door de pagina opnieuw te laden zonder netwerkverbinding!)
- Ruimt de eerder geprecachede items op wanneer de cache naam wordt bijgewerkt
- Onderschept netwerkverzoeken en stuurt een antwoord uit de cache terug als dat beschikbaar is
- Als er geen antwoord in de cache is, haalt het antwoord op van het netwerk en voegt het toe aan de cache voor toekomstig gebruik

Handige bronnen:

- [Web app service worker](#)
- [Basic service worker example](#)



```
self.addEventListener('install', function(event) {
  event.waitUntil(
    caches.open('sw-cache').then(function(cache) {
      return cache.add('index.html');
    })
  );
});

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

De bovenstaande bestanden doen het minimale om je service worker te laten werken. Zie de link naar het example, voor een mooie basis code om mee te starten.