

Keuzedeel: Mobile Application Development (K0497)

Opdracht 4: PWA & Sensors For The Web

Tijdens deze opdracht maak je een PWA (progressive web app).

Binnen deze app maak je gebruik van de Generic Sensor API.

Dat doe je door gebruik te maken van een combinatie van verschillende technieken en diensten.

Bekijk het online voorbeeld om een beeld te krijgen van wat de bedoeling is:

<https://glu-mad-pwa-sensor-onsen.herokuapp.com/>

De sourcecode van dit voorbeeld is terug te vinden op GitHub:

<https://github.com/martijkunstman/GLU-MAD-PWA-sensor-onsen>

Progressive web apps vervangen steeds meer traditionele native apps.

Ze zijn gebaseerd op web technologieën en kunnen net zoals een native app worden toegevoegd aan het startscherm. Dit maakt ze erg geschikt om mee aan de slag te gaan binnen MAD. Ze kunnen namelijk met bestaande kennis van HTML/CSS/JS worden gemaakt.

Door de **Generic Sensor API** is het mogelijk geworden de sensoren van mobile-devices óók te gebruiken zonder dat daarvoor een native app noodzakelijk is.

Deze twee ontwikkelingen zijn zeer interessant met betrekking tot MAD, omdat de noodzaak voor het ontwikkelen van native apps hierdoor steeds minder wordt.

PWAs

Progressive web applications (PWAs) are a type of application software delivered through the web, built using common web technologies including HTML, CSS and JavaScript. They are intended to work on any platform that uses a standards-compliant browser. Functionality includes working offline, push notifications, and device hardware access, enabling creating user experiences similar to native applications on desktop and mobile devices. Since they are a type of webpage or website known as a web application, there is no requirement for developers or users to install the web apps via digital distribution systems like Apple App Store or Google Play.

Source: https://en.wikipedia.org/wiki/Progressive_web_applications

Sensors for the Web

Today, sensor data is used in many native applications to enable use cases such as immersive gaming, fitness tracking, and augmented or virtual reality. Wouldn't it be cool to bridge the gap between native and web applications? The Generic Sensor API, For The Web!

At the time of writing, Chrome supports several sensors that you can experiment with.

Motion sensors:

Accelerometer

Gyroscope

LinearAccelerationSensor

AbsoluteOrientationSensor

RelativeOrientationSensor

Environmental sensors:

AmbientLightSensor

Magnetometer

Source: <https://developers.google.com/web/updates/2017/09/sensors-for-the-web>

Generic Sensor

Increasingly, sensor data is used in application development to enable new use cases such as geolocation, counting steps or head-tracking. This is especially true on mobile devices where new sensors are added regularly.

Exposing sensor data to the Web has so far been both slow-paced and ad-hoc. Few sensors are already exposed to the Web. When they are, it is often in ways that limit their possible use cases (for example by exposing abstractions that are too high-level and which don't perform well enough). APIs also vary greatly from one sensor to the next which increases the cognitive burden of Web application developers and slows development.

Source: <https://www.w3.org/TR/generic-sensor/>

Hoe ga je te werk?

Voor het hosten van je PWA kun je gebruik maken van diverse gratis online services. Het is daarbij echter wel van belang te kiezen voor een optie waarbij https mogelijk is. Dat is noodzakelijk voor zowel de Sensor API als PWA.

Je kunt bijvoorbeeld gebruik maken van:

<https://firebase.google.com/>

Of

<https://www.heroku.com/>

In dit voorbeeld is gebruik gemaakt van **Heroku**.

Na het aanmaken van een account kun je binnen Heroku een nieuwe app aanmaken en die koppelen aan een GitHub repository. Daarna kun je met een druk op de knop je PWA deployen.

Voor de basis van de app is gebruik gemaakt van het **OnsenUI** framework.

<https://onsen.io/>

Met dit framework kun je de look and feel van een native app nabootsen.

Via de playground kun je snel en makkelijk voorbeelden bekijken van de mogelijkheden én bijbehorende code.

<https://onsen.io/playground/>

Om te starten met een lege template die je zelf kunt gaan vullen kun je gebruik maken van deze tutorial:

<https://onsen.io/v2/guide/tutorial.html#tutorial>

Volg daarna deze tutorial om een PWA te genereren:

<https://onsen.io/v2/guide/pwa/tutorial.html>

Om binnen je app gebruik te maken van de sensoren van je mobile-device kun je gebruik maken van de Generic Sensor API:

<https://www.w3.org/TR/generic-sensor/>

Voor het gebruik van de microfoon en camera van je mobile-device kun je veel informatie (en code) vinden op deze pagina:

<https://www.html5rocks.com/en/tutorials/getusermedia/intro/>

Vergeet binnen je app niet node.js te installeren en de bijbehorende index.js

Repository

<https://github.com/martijnkunstman/GLU-MAD-PWA-sensor-onsen>

Resultaat

<https://glu-mad-pwa-sensor-onsen.herokuapp.com/>

Wat moet je doen?

Tijdens deze opdracht werk je individueel aan het maken van een PWA waarbinnen je gebruik maakt van de Generic Sensor API. Je bepaald zelf hoe je dat aan wilt pakken. Je kunt de werkwijze volgen zoals die in deze opdracht wordt gesuggereerd. De inhoud van de app is ondergeschikt aan het feit of je de opdracht tot een goed einde weet te brengen. Een goed einde is in dit geval: een PWA die werkt op je mobile device en die kan worden opgestart door een snelkoppeling op je startscherm.

Tijd, hoeveel krijg je?

Je krijgt voor deze opdracht 2 weken de tijd. In totaal is dat 8 uur.

Waarom doen we dit?

De uitwerking van deze opdracht kun je gebruiken voor je examen.

Tijdens het CGI kun je bijvoorbeeld deze app demonstreren.

Binnen deze opdracht maak je kennis met uiteenlopende technieken en diensten die relevant zijn voor MAD.