OBERON

React en React Native voor websites en apps

HANS-PETER HARMSEN HEEFT DIT GESCHREVEN IN • **APRIL 2017**

Deze whitepaper is bedoeld voor product owners en beslissers. Hij gaat over React, een JavaScript library voor het bouwen van gebruikersinterfaces. We leggen uit wat React is en waarom je er betere websites en apps mee kunt maken.

Inleiding

React is een Javascript library voor het bouwen van user interfaces. React bestaat in twee smaken. De eerste, React Web, is bedoeld voor de website ontwikkeling en wordt eigenlijk gewoon React genoemd. De tweede is React Native, speciaal voor de ontwikkeling van mobiele apps. React is in eerste instantie ontwikkeld door Facebook en wordt nu onderhouden door Facebook en Instagram samen met een gemeenschap van onafhankelijke software ontwikkelaars en bedrijven. Bekende websites op basis van React zijn Netflix, AirBnB, New York Times, Yahoo Mail en de webversie van WhatsApp.

De websites van AirBnB, New York Times, Netflix en WhatsApp zijn gemaakt met React

React voor webontwikkeling

Bij Oberon voor React gekozen voor de ontwikkeling van websites, web-applicaties en apps. Dit heeft drie redenen: 1) omdat React een elegante manier biedt om web-applicaties te ontwikkelen; 2) omdat de code er netjes en onderhoudbaar van wordt en 3) omdat de producten die we met React ontwikkeld hebben een veel natuurlijkere 'feel' hebben.

REDUX

React werkt heel lekker samen met Redux; een web framework dat in feite los staat van React maar beiden worden wel vaak als één gezien. Redux werkt als volgt:
Als er iets verandert op een pagina, bijvoorbeeld om dat er data is ingeladen of omdat de gebruiker ergens op geklikt heeft, dan past de applicatie-code alleen de onderliggende data aan. Redux zorgt er vervolgens voor dat deze aanpassingen worden doorgegeven aan de UI. Die UI wordt gevormd door React. React hertekent alleen die delen van de pagina die veranderd zijn. Deze, oogenschijnlijk omslachtige werkwijze, zorgt ervoor dat de precieze 'state' van de applicatie maar op één plek wordt bijgehouden, namelijk in Redux, en dat de UI zich daar automatisch op aanpast.

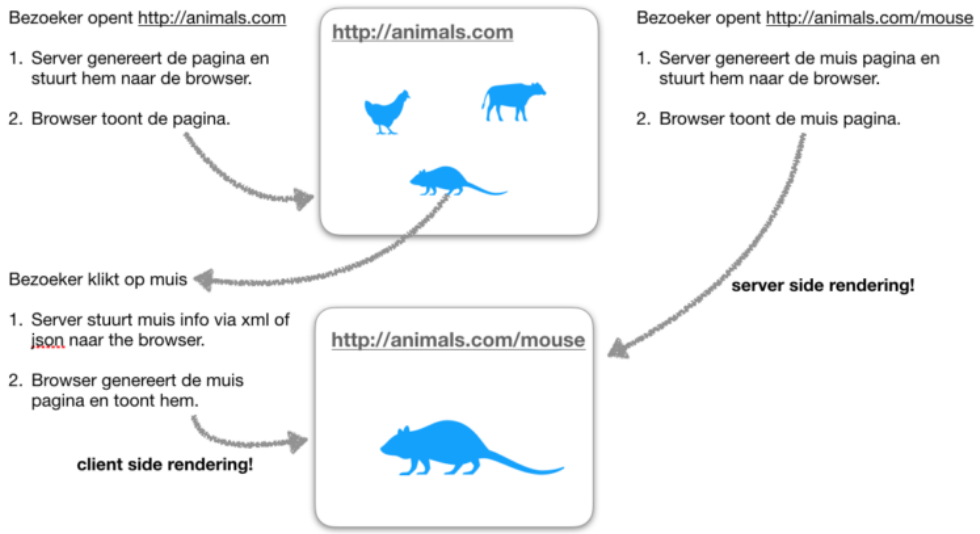
Voor de gebruiker geeft dit een veel snellere website en een vloeiende gebruikerservaring. Ideaal dus voor voor Single Page Applications.

De actie en reactie cirkel van Redux

SERVER SIDE RENDERING

Een ander voordeel is server side rendering, ook wel isomorphic loading genoemd. Bij een normale moderne webapplicatie wordt de opmaak van de pagina als html ingelezen en komt de data apart van de server vanuit een api. Dit werkt mooi maar heeft tot resultaat dat het tonen van de pagina wat vertraagd wordt. Eerst moet tenslotte de html ingelezen worden, dan de data en tenslotte moet de browser dit combineren tot een volledige pagina.

Bij server side rendering gebeurt dit direct al op de server. De pagina wordt, al geheel samengesteld, in een keer naar de browser gestuurd en kan meteen getoond worden. Verdere aanpassingen gebeuren wel lokaal in de browser zodat er niet steeds een verzoek naar de server nodig is voor alles wat de gebruiker doet. In feite draait dezelfde code dus zowel op de server als in de browser. En dat merk je in de snelheid!



Schematische weergave van client side rendering vs. server side rendering

GEOPTIMALISEERD VOOR SEARCH ENGINES

Een nadeel van SPA's (Single Page Applications) is dat ze SEO technisch niet optimaal zijn. Search engines werken op basis van het ophalen van pagina's op basis van URL's; de pagina's achter die URL's worden één voor één geïndexeerd. Een SPA heeft echter maar één pagina; de rest van de applicatie wordt gegenereerd met behulp van JavaScript in de browser. Search engines voeren maar beperkt JavaScript uit dus de search engine zal vooral de eerste pagina zien.

Server side rendering lost dit op. Als een search engine een URL uit de applicatie ophaalt, dan zorgt de server er voor dat de pagina netjes opgebouwd wordt.

Technisch werkt dit zo dat dezelfde JavaScript code die in de browser draait om pagina's op te bouwen ook op de server draait. Het maakt dus niet meer uit of een pagina in de browser wordt gegenereerd (voor de vloeiende gebruikerservaring) of op de server (voor search engines of als je als gebruiker direct naar een onderliggende url springt).

Een mooi voorbeeld van een website waarin dit allemaal samen komt is idfa.nl.

React Native voor mobiele apps

React maakt het mogelijk om een app in één keer te ontwikkelen voor iOS en Android tegelijk. Voor de ontwikkeling van mobiele apps bestaan al veel langer hybride systemen, zoals Phone Gap, maar daar zijn wij als Oberon altijd ver weg van gebleven. Dergelijke platforms zijn eigenlijk niet veel anders dan een lokale website met een app-schilletje eromheen; slecht aanpasbaar aan het specifieke besturingsysteem en traag in de bediening. Om die reden kozen wij tot nu toe altijd om losse 'native' apps te ontwikkelen voor iOS en voor Android. Dit betekende dus wel altijd dubbel werk. Alleen het design kon (grotendeels) gedeeld worden.

De apps van Instagram, Baidu, Facebook en AirBnB zijn gemaakt met React Native

Wat maakt React Native dan anders? De naam zegt het al React Native zet de JavaScript code om naar 'native' iOS en Android componenten. Daarnaast is het mogelijk om vanuit React eigen native componenten te gebruiken voor onderdelen waarin React niet voorziet zoals het uitlezen van de camera of andere hardware. Door deze combinatie met 'native' componenten krijgen apps de kracht van 'native' terwijl ze toch maar één keer ontwikkeld hoeven te worden.

En dit voordeel van maar één keer schrijven zet zich natuurlijk voort in het onderhoud van de app. Ook aanpassingen hoeven namelijk maar één keer gedaan te worden.

De combinatie en React en React Native

De combinatie van React en React Native geeft ook eigen voordelen. Ten eerste leidt dit tot een eenduidige manier van ontwikkelen. Alle code wordt op dezelfde manier opgezet in dezelfde programmeertaal, namelijk JavaScript (officieel: ECMAScript 6). Ten tweede kunnen we hiermee de code die de bedrijfslogica regelt, hergebruiken. We kunnen precies dezelfde code gebruiken voor de website als voor de bijbehorende app. Alleen de user-interface code is echt anders.

Windows Phone

We hebben het tot nu toe alleen gehad over iOS en Android apps. Maar React Native kan ook worden gebruikt om apps te maken voor Windows Phone. Dit wordt nog niet officieel ondersteund maar er zijn al wel implementaties voor. Wij hebben zelfs al een Windows versie ontwikkeld voor een van de apps die we bij Oberon maken: de Health Care Professionals app voor Nutricia. Omdat Windows Phone support nog niet officieel is, is het vrij lastig om dit goed voor elkaar te krijgen, maar het is gelukt. Onze eerste Windows Phone app vanuit React Native staat binnenkort in de Microsoft Store.

Conclusie

React Web is een goede manier om krachtige websites en web-applications te maken. Het maakt het ontwikkelaars makkelijk om een nette onderhoudbare code te schrijven die een prettige gebruikerservaring combineert met goede SEO mogelijkheden.

Haar broertje React Native gebruiken we voor het ontwikkelen van iOS en Android apps met de mogelijkheden en gebruikservaring van native apps maar zonder de overhead van het bouwen van twee aparte apps.

HANS-PETER HARMSEN
MANAGEMENT

Oprichter en managing director; verantwoordelijk voor grote accounts en strategie.

E-MAIL: **HPH@OBERON.NL**
TELEFOON: **+31 654 337 275**

OBERON

“We maken samen met onze klanten betere online producten, zowel voor het web als in mobiele apps”

MEER INFORMATIE OP **OBERON.NL**