

Inhoud

1	Vijf W's en een H – React	1
	Inleiding	2
	Voor wie is React bedoeld?	2
	Wanneer is React ontwikkeld?	4
	Waarom is React ontwikkeld?	5
	Wat is React?	8
	Een library	8
	Een framework	9
	Inversion of control (IoC)	10
	JavaScript-library's en -frameworks	11
	Veelgebruikte JavaScript-library's	12
	Veelgebruikte JavaScript-frameworks	13
	JavaScript-frameworks/library's en trage opbouw	14
	Nuttige tools voor veel toepassingen	15
	Hoezo React en hoe gaat u React zich eigen maken?	15
	Hoezo... React?	16
	Hoe gaat u React zich eigen maken?	19
	Verandering in uw werk- en denkwijze	20
	Samengevat	20
2	Bouwstijlen	21
	Inleiding	22
	Bouwstijlen in programmeren	22
	Een fundament leggen: het DOM	22
	Het verschil tussen virtual DOM en DOM	23
	Imperatief programmeren	31
	Declaratief programmeren	32
	Functioneel programmeren in het kort	36
	JavaScript	36
	JavaScript is de basis	38
	JavaScript Runtime Environment	38

	Functioneel programmeren	42
	Veelbelovende kenmerken van functioneel programmeren	43
	First-class citizens	45
	Immutability	55
	Purity	61
	Data transformers	68
	Recursie	70
	Composition	74
	Closures	76
	Destructuring	77
	Samenvatting	80
3	Bouwsystemen	81
	Inleiding	82
	Node.js	82
	Inleiding tot create-react-app	85
	Create-react-app	86
	Installatieloos uitvoeren van de create-react-app toolchain	86
	Verkennen van de mappenstructuur	90
	De map .git	91
	De map node_modules	92
	De map public	92
	De map src	93
	Het bestand .gitignore	93
	Het bestand yarn.lock	94
	Het bestand package.json	94
	De ontwikkelingsserver starten	96
	Samenvatting	97
4	Bouwstenen	99
	Inleiding	100
	Belangrijke begrippen	100
	Drie soorten applicatievormen	106
	Informatievorm	106
	Verwerkvorm	107
	Bewerkvorm	107
	Grondregel: één enkele verantwoordelijkheid	108
	ReactDOM versus ReactComponent	109
	ReactDOM	109
	ReactComponent	110
	Wat maakt het verschil?	111
	Samenvatting	112

5	Bouwproject 1	113
	Bouw uw eerste React-app	114
	Visual Studio (VS) Code vanaf de prompt starten	114
	Stap 1. Splits de appuitvoer op in componenten	116
	Naamconventie voor componenten	117
	Opbouw van de mappenstructuur	119
	Stap 2. Bouw de UI met passieve componenten	120
	Minimale opmaak eerst	120
	Kies een fontstapel	121
	Stap 3. Beslis welke gegevens actief veranderen en in welke componenten ze voorkomen	128
	Stap 4. Bouw de componenten om: laat ze eerst met een vaste gegevensset werken	128
	Het datamodel	128
	Stap 5. Voeg een variabele gegevensstroom van ouder- naar kindcomponenten toe	131
	Wanneer splitst u een component in meerdere componenten op	134
	Stap 5. Voeg een variabele gegevensstroom van ouder- naar kindcomponenten toe	141
	Gegevens van ouder- naar kindklassecomponent met behulp van props	141
	Gegevens van ouder- naar kindfunctiecomponent met behulp van props	143
	Stap 6. Voeg een gegevensstroom van kind- naar oudercomponenten toe	146
	Gegevens van kind-naarouderfunctiecomponent met behulp van callbacks	146
	Stap 7. Bouw een communicatiecomponent die externe koppelingen kan leggen	147
	Een eigen hook coderen	147
	Samenvatting	151
6	Bouwproject 2	153
	Opzet en installatie	154
	create-react-app	154
	Ontwikkelhulpmiddelen	156
	JSX: JavaScript + XML	157
	Componenten	158
	Klassecomponenten	159
	Simpel componenten	161
	Props	163
	State	168
	Formuliergegevens	172
	API-gegevens ophalen	177
	Een React-app bouwen en uitrollen	179
	Samenvatting	181

7	Bouwproject 3	183
	Een CRUD-applicatie in React met hooks	184
	Doelstellingen	185
	Creëer een React-applicatie met React v16.8.4	185
	Eerste installatie	186
	State versus hookstate	187
	De view opzetten	188
	Een nieuwe gebruiker toevoegen	192
	Een gebruiker verwijderen	198
	Een gebruiker bijwerken	199
	Het formulier EditUserForm.js	201
	De effecthook gebruiken	203
	Samenvatting	205
8	Hulpprogramma's en hulpmiddelen	207
	Mogelijkheden van de browser	208
	Hoe opent u de DevTools in uw browser?	208
	Tabblad Elements	209
	Tabblad Console	209
	Tabblad Network	209
	Mogelijkheden van de React DevTools	209
	Tabblad Components	210
	Rechtstreeks bewerken van state en props	210
	Boomslingeren	210
	Componentenfilter	210
	Weergegeven door	211
	Boomeigenaar	211
	Snel schakelen	211
	Hulpmiddelen	212
	Tabblad Profiler	212
	Commitgrafiek	213
	Componentgrafiek	213
	Profiler starten	213
	Rendered at	214
	Optimaliseren met de React DevTools Profiler	215
	Context creëren	216
	Context leveren	218
	Context consumeren in klassen	219
	Context consumeren in functionele componenten en hooks	219
	Nulmeting van de applicatie	221
	React.memo	222
	Conclusie	225

	Nawoord	226
	Ja, en nu?	226
A	React.memo	229
	Wat moet u weten over React.memo?	230

Vijf W's en een H – React

In acht hoofdstukken pak ik met u de beginselen van React aan die leiden tot het maken van mooie, eigen webapplicaties. Het creëren van herbruikbare componenten, webapplicaties coderen en het structureren en optimaliseren van uw applicatie: het komt allemaal aan bod.

In de eerste hoofdstukken leert u de basis van React kennen, dus maakt het geen verschil of u al de nodige kennis heeft opgedaan. De basis blijft hetzelfde en wanneer u deze basiskennis al beheerst, dan kunt u zelf bepalen om verder te gaan vanaf één van de resterende hoofdstukken.

In dit hoofdstuk richt u zich op:

De historie en het ontstaan van React.

Voor wie het bedoeld is en waarom.

Wat u nodig heeft om React u eigen te maken.

Inleiding

‘5W+H’ staat voor een benadering die helpt bij het duidelijk krijgen van wat ik u na het bestuderen van dit boek wil meegeven. Door het beantwoorden van een aantal vragen krijgt u een goed beeld van hoe React eruit ziet. Met deze aanpak help ik u na te denken over waarom, wat, wie, wanneer, waar en hoe. Ik begeleid u op een ontdekkingsreis in de kunst van applicatieontwikkeling in React.

Voor wie is React bedoeld?

React is moeiteloos te leren en heeft een lage leercurve in vergelijking met andere JavaScript-frameworks en bibliotheken zoals Vue.js, Angular en Ember. Een en ander op voorwaarde dat u openstaat voor veranderingen in uw manier van werken en denken.



Frameworks

Er zijn veel populaire JavaScript-frameworks beschikbaar:

- Vue.js is een open-source JavaScript-framework voor het bouwen van gebruikersinterfaces en *single page applications*. De eerste release (2014) is ontwikkeld door Evan You.
 - Angular 2+, AngularJS zijn Java- en TypeScript-gebaseerde open-source webapplicatieframeworks. De eerste release (2010) is ontwikkeld door het Angular Team van Google.
 - Ember.js is een open-source JavaScript-webframework. De eerste release (2011) is ontwikkeld door Yehuda Katz.
-

Wat stopt u in uw rugzak als u de komende tijd de React-wereld rondreist? React is te leren met wat kennis van en vaardigheid in JavaScript of een andere programmeer- of scriptingtaal. De basis van JavaScript wordt vaak aangeduid als JavaScript ECMAScript 5, beter bekend als ES5. React maakt gebruik JavaScript, meestal gebaseerd op JavaScript ES6 en verder (ES7, ES8, ES9).



JavaScript

JavaScript is een krachtige, geïnterpreteerde programmeertaal die voldoet aan de ECMAScript-specificatie. De eerste release is in 1995 ontwikkeld door Brendan Eich.

U moet in ieder geval de werking van onderstaande JavaScript-onderwerpen kennen:

- Variabelen en typen
- Objecten en arrays
- Klassen en functies
- Lussen (*loops*) en voorwaardelijke statements (*conditionals*)

Verder hebt u ervaring nodig met het opzetten van een webpagina met HTML en CSS:

- HTML (Hypertext Markup Language) is dé standaard opmaaktaal voor het maken van webpagina's en webapplicaties. Het begon als het prototype van het ENQUIRE-systeem voor CERN om documenten te gebruiken en te delen. HTML is voorgesteld in 1989 en de eerste release is in 1993 ontwikkeld door Tim Berners-Lee.
- CSS (Cascading Style Sheets) is een conventie die de vormgeving beschrijft voor een document geschreven in een opmaaktaal zoals HTML. De eerste release (1996) is ontwikkeld door Håkon Wium Lie en Bert Bos.



JavaScript

Heeft u al de nodige ervaring met JavaScript opgedaan maar nog niet met de recente mogelijkheden die in de afgelopen paar jaren zijn geïntroduceerd? Lees dan vooral door! In de volgende hoofdstukken pak ik dit met u op, en langzaam maar zeker stoom ik u klaar voor alles wat JavaScript tegenwoordig aan mogelijkheden biedt.

In applicatieontwikkeling worden veel Engelse woorden en begrippen gebruikt. In dit boek zal ik waar nodig het Engelse begrip tussen haakjes achter het Nederlandse woord plaatsen en stilletjes overgaan op het gebruiken van het Engelse woord of begrip. Reden is dat de meest actuele informatie en documentatie in het Engels is beschreven en het dus noodzakelijk is deze woorden/begrippen te kennen. In code is dit niet anders en wereldwijd aanvaard.

Nu u weet welke kennis en kunde er in uw rugzak thuishoren, nodig ik u uit om op reis te gaan in de wondere wereld die React heet.

Wanneer is React ontwikkeld?

We reizen terug in de tijd, het is dinsdag 9 februari 2010, ongeveer 4 uur in de middag. Marcel Georgés Laverdet is een Facebook-gebruiker van het eerste uur.



Facebook

Facebook, Inc. is een Amerikaanse onderneming die online social media en social networking services levert. Opgericht door Mark Zuckerberg, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz en Chris Hughes. Eerste release 2004, ontwikkeld door Mark Zuckerberg.

Goed bevriend met Mark Zuckerberg en werkzaam als applicatieontwikkelaar bij Facebook introduceert hij XHP. XHP was bedoeld als aanvulling op de front-end PHP-omgeving van Facebook.

XHP maakte het mogelijk om met XML herbruikbare HTML-elementen te ontwikkelen, kortom samengestelde componenten. Facebook maakt nog altijd gebruik van XHP.

Geïnspireerd door XHP begon Jordan Walke (ook een applicatieontwikkelaar bij Facebook) in september 2011 met het ontwikkelen van FaxJS, het eerste prototype van React. Dit markeert de geboorte van ReactJS.

Waarom is React ontwikkeld?

In 2011 kregen de ontwikkelaars van Facebook te maken met problemen bij het onderhoud van de code. Omdat de Facebook Ads-app steeds meer functionaliteiten kreeg, had het team meer mensen nodig om de applicatie vlekkeloos te laten draaien. Het groeiende aantal teamleden en applicatiefunctaliteiten remde het bedrijf af. De applicatie was moeilijk te hanteren, omdat het team te maken kreeg met een steeds grotere hoeveelheid aan wijzigingen. Na verloop van tijd konden de ontwikkelaars van Facebook de stroom van wijzigingen niet meer bijhouden. Hun code schreeuwde om grote verbeteringen, om zo efficiënter en robuus-

ter te worden. Ze hadden wel het juiste model, maar ze moesten iets doen aan de gebruikerservaring. Jordan Walke had inmiddels een eerste prototype gebouwd in de vorm van FaxJS.

We schrijven inmiddels 9 april 2012. Krantenkoppen melden “Instagram overgenomen door Facebook”.



Instagram

Instagram (ook bekend als IG of Insta) is een dienst voor het delen van foto's en video's op sociale netwerken van Facebook, Inc., opgericht in 2010 door Kevin Systrom en Mike Krieger in San Francisco, Californië.

Instagram wilde graag de nieuwe veelbelovende technologie van Facebook gebruiken. Hierdoor zag Facebook zich genoodzaakt om React los te koppelen van Facebook en de broncode klaar te maken voor wereldwijde vrijgave in de opensourcegemeenschap. Het grootste deel hiervan werd gedaan door Pete Hunt. 29 mei 2013 was het zover: de lancering van ReactJS was een feit. De rest van dat jaar nam de populariteit sterk toe.

In 2014 begint React door te dringen tot potentiële gebruikers. Door zich op het robuuste karakter van React te richten, wilden de ontwikkelaars bedrijven als Netflix aanspreken, wat lukt. Acties worden ondernomen om de gemeenschap verder uit te bouwen en om 'haters te hervormen tot fanatieke voorstanders'.



Netflix

Netflix, Inc. is een Amerikaanse mediadienstverlener, opgericht in 1997 door Reed Hastings en Marc Randolph in Scotts Valley, Californië.

Eind januari 2015 onthult Facebook de eerste versie van React Native, bedoeld voor mobiele apps; het wordt enthousiast ontvangen. Netflix en Airbnb gebruiken ReactJS voor een groot gedeelte van zowel de voorkant als de achterkant van hun web-applicaties.



Airbnb

Airbnb, Inc. is een particulier wereldwijd opererend bedrijf dat een online marktplaats en horecadienst exploiteert die toegankelijk is via haar websites en mobiele apps, opgericht in 2008 door Brian Chesky, Joe Gebbia en Nathan Blecharczyk in San Francisco, Californië.

Op 28 september 2016 worden de eerste 50.000 GitHub stars gehaald (zie reactjs.org/blog/2016/09/28/our-first-50000-stars.html). Tot de dag van vandaag zet de ontwikkeling sterk door en de populariteit van React bereikt keer op keer recordhoogtes.



GitHub

GitHub is een web-based hostingservice voor versiebeheer met behulp van Git, opgericht door Tom Preston-Werner, Chris Wanstrath, P. J. Hyett en Scott Chacon in San Francisco, Californië.

Hiermee bereikt u uw volgende tussenstop, maar er is nog veel meer te leren over React.

Wat is React?

React is een JavaScript-bibliotheek (*library*) voor het bouwen van samenstelbare gebruikersinterfaces (*UI*). Het moedigt de ontwikkeling van herbruikbare UI-componenten aan die gegevens (*data*) weergeven die na verloop veranderen. Maar de vraag rijst: wat zijn frameworks en library's?

Om te beginnen is het belangrijk om twee begrippen uit elkaar te houden. Library (bibliotheek) en framework (raamwerk) worden vaak door elkaar gebruikt, wat strikt genomen verkeerd is. Ze overlappen elkaar weliswaar op veel gebieden, maar toch zijn er fundamentele verschillen.

Een library

Een library is een programmeerbibliotheek, of het nu om een klassen-, componenten- of een andere bibliotheek gaat. Programmeerbibliotheken hebben altijd subprogramma's met hulpfuncties die het programmeren vereenvoudigen. In vergelijking met een framework is een bibliotheek meer ontwikkeld voor een specifiek gebruik en heeft hiervoor op elkaar afgestemde functies. De JavaScript-library D3.js (zie d3js.org) dient bijvoorbeeld voor data-visualisatie – hiermee kunnen zowel kleine tabellen, diagrammen en statistieken als complexe grafische voorstellingen worden geïmplementeerd (inclusief animaties en interacties). Library's werken altijd met een geïntegreerd softwareprogramma dat de betreffende functies van een programmeerbibliotheek gebruikt als er een bepaalde functie nodig is. Daarom kunnen library's alleen binnen een programma werken, en niet zelfstandig.



D3.js

D3.js (Data-Driven Documents) is een JavaScript-bibliotheek voor het produceren van dynamische, interactieve datavisualisaties in webbrowsers. De eerste release is in 2011 ontwikkeld door Mike Bostock, Jason Davies, Jeffrey Heer en Vadim Ogievetsky.

Een framework

Een *framework* is ook geen autonoom programma, maar eerder een speciale vorm van een klassenbibliotheek. Een framework stelt de softwarearchitectuur (dus de basisstructuur) van een applicatie voor en bepaalt in feite het ontwikkelingsproces. Frameworks hebben bepaalde ontwerppatronen met verschillende functies (vaak in de vorm van meerdere bibliotheken) en dienen voor de ontwikkeling van nieuwe, onafhankelijke applicaties. Een voorbeeld is Zend Framework voor PHP dat onder andere wordt gebruikt in de webwinkelsoftware van Magento en de webanalysetool Matomo, beter bekend als Piwik.



Zend Framework

Zend Framework (ZF) is een open-source, objectgeoriënteerd webapplicatieframework geïmplementeerd in PHP 7. De eerste release (2006) is ontwikkeld door Zend Technologies.



Magento

Magento is een open-source e-commerce-platform geschreven in PHP. De eerste release (2008) is ontwikkeld door Varien in Culver City, Californië.

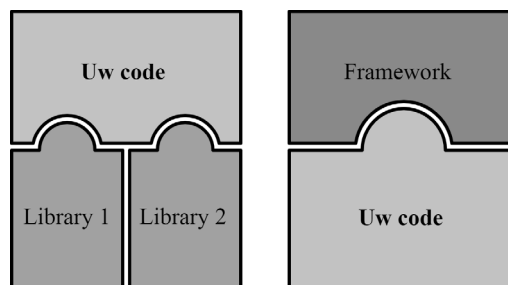


Matomo

Matomo, voorheen Piwik, is een gratis en open-sourceapplicatie voor *web analytics*. De eerste release is in 2007 gemaakt door een team van internationale ontwikkelaars.

Inversion of control (IoC)

Een ander verschil tussen een framework en een bibliotheek is de wijze van besturing. Bij bibliotheken hebben applicatieontwikkelaars toegang tot de code via de programmeerinterface van de software. Frameworks voeren daarentegen een zogeheten omkering van de besturing uit (*inversion of control*): de code is ingesloten in vastgestelde structuren van het framework en kan worden opgeroepen zodra deze nodig is. Samenvattend kan worden gezegd dat bibliotheken worden opgeroepen door het programma, terwijl een framework de programmacriteria stelt.



Afbeelding 1.1 *Inversion of control.*

JavaScript-library's en -frameworks

JavaScript (afgekort JS) is een scripttaal waarmee dynamische HTML-content in de webbrowser interactief kan worden gemaakt. De taal kan worden gebruikt voor zowel objectgeoriënteerd als procedureel en functioneel programmeren. JavaScript bevat voor-gedefinieerde objecten (dat wil zeggen dataelementen met speciale eigenschappen, methodes of functies) voor de toegang tot een website, maar die zijn soms lastig in het gebruik. Inmiddels is veel nuttige informatie verzameld in verschillende JavaScript-bibliotheken (in het Engels: *JavaScript libraries*) en -frameworks. Dit zijn tools die het programmeren vereenvoudigen. Ik bespreek hier de beste en bekendste JavaScript-tools.

De kern van JavaScript is een tamelijk eenvoudige programmeertaal die bijzonder geschikt is voor gebruik binnen een browser. Veel applicatieontwikkelaars hebben problemen met de interface van hun website – of liever gezegd het DOM (Document Object Model), een standaard die wordt bijgehouden door de WHATWG-groep (html.spec.whatwg.org/dev). Hier komen JavaScript-frameworks en -library's van pas: ze vereenvoudigen het werk van ontwikkelaars op verschillende gebieden van het programmeren. De JavaScript-extensions die hieronder worden voorgesteld, kunnen allemaal gratis worden gedownload.



WHATWG

De WHATWG is een groep van geïnteresseerden in de ontwikkeling van HTML en bijbehorende technologieën. Het is opgericht door Apple Inc., Mozilla Foundation en Opera Software in 2004 als reactie op de trage ontwikkeling van World Wide Web Consortium (W3C)-webstandaarden en het besluit van het W3C om HTML op te geven en te vervangen door XML-gebaseerde technologieën.

Veelgebruikte JavaScript-library's

JavaScript-library's zijn verzamelingen herbruikbare code waarmee bepaalde eigenschappen en functies voor een website kunnen worden toegevoegd. jQuery is de bekendste JavaScript-library en biedt veel nuttige functies, hoewel ook andere bibliotheken behulpzaam kunnen zijn.

jQuery

De uitgebreide bibliotheek jQuery (**jquery.com**) is de meestgebruikte JavaScript-library, onder meer omdat u hiermee gemakkelijk cross-browser jQuery-code kunt schrijven en omdat er veel plug-ins voor bestaan. Deze opensourcebibliotheek maakt deel uit van veel contentmanagementsystemen, zoals WordPress, Drupal en Joomla!. jQuery dient als interface met het DOM en biedt de volgende functies:

- Eenvoudige selectie van website-elementen (vergelijkbaar met CSS3-selectors).
- Mogelijkheid om geselecteerde website-elementen te veranderen (zoals positie, kleur enzovoort).
- Event-systeem: website-elementen reageren op usernavigatie (muisklikken, toetsenbordinput enzovoort).
- Eenvoudige implementatie van animaties en effecten.
- Naadloze interactie tussen invoer en serverdata met Ajax-requests (zoals bij de autocompletefunctie).

jQuery UI

jQuery UI (**jqueryui.com**) is een gratis uitbreiding voor jQuery. Hiermee kunt u een gebruikersinterface (in het Engels: *user interface* of afgekort UI) vormgeven en implementeren van bijvoorbeeld webapplicaties. De focus ligt op het eenvoudig ontwerpen van effecten en interacties. jQuery UI heeft als functie het implementeren van interactiemogelijkheden (onder andere *drag and drop*, inzoomen en uitzoomen), maar ook animaties, effecten en

widgets (autocomplete, sliders, datumkiezer enzovoort). Met de grafische editor ThemeRoller kunnen eigen thema's worden gemaakt of bestaande thema's worden gebruikt en aangepast. Door de modulaire structuur hoeven alleen de noodzakelijke componenten te worden geïmplementeerd.

Dojo Toolkit

Dojo Toolkit (dojotoolkit.org) is vooral geschikt voor de implementatie van webapplicaties en dynamische webcontent. De toolkit heeft een grote diversiteit aan functies. De belangrijkste bestanddelen zijn:

- Dojo (het hoofdprogramma): snel en eenvoudig gebruik van DOM, Ajax, JavaScript, events enzovoort.
- Dijit: een soort module voor grafische gebruikersinterfaces, widgets, lay-outs, effecten en animaties (vergelijkbaar met jQuery UI).

React

React (facebook.github.io/react/) werd in 2011 voor het eerst gebruikt in een Facebook-newsfeed. In 2013 werd het vervolgens gepubliceerd op opensourcebasis. Het is een JavaScript-library waarmee u gebruikersinterfaces kunt maken. Het bijzondere van deze bibliotheek is dat website-elementen (*views*) niet alleen op de client kunnen worden gerenderd, maar ook op de server, wat een sterkere rendering mogelijk maakt. Dit kan door het gebruik van een *virtual DOM*, waarmee ook het testen van webapplicaties eenvoudiger wordt.

Veelgebruikte JavaScript-frameworks

Er zijn aanzienlijk meer JavaScript-library's dan frameworks. Frameworks zijn in het bijzonder geschikt voor complexe webapplicaties. Wanneer ontwikkelaars openstaan voor de concepten en specificaties van de betreffende frameworks, kunnen ze er

heel effectief mee werken. Twee zeer uitgebreide en vaakgebruikte tools zijn AngularJS en Ember.js.

AngularJS

Dit framework, dat van Google afkomstig is, heeft veruit de grootste community van alle JavaScript-frameworks. Angular (angular.io) kan worden gebruikt om *single-page-webapplicaties* te maken (webapps die uit slechts één HTML-document bestaan). Door het MVVM (*model-view-viewmodel*) kunnen webapplicaties worden ontwikkeld die zeer goed zijn afgestemd op interactie met de gebruiker. Daarbij rendert AngularJS de webapplicatie aan de clientzijde. Het framework is gebaseerd op jQuery Lite, een afgeslankte versie van jQuery.

Ember.js

Ook Ember.js (emberjs.com) is een client-side framework dat wordt gebruikt voor het implementeren van single-page webapplicaties. In tegenstelling tot Angular is het gebaseerd op het MVC-model (*model-view-controller*). Een ander onderscheidend kenmerk is dat de makers van Ember.js de community intensief betrekken bij de ontwikkelprocessen van het framework en met hen discussiëren over belangrijke veranderingen aan het framework, voordat ze worden geïmplementeerd. Ember.js omschrijft zichzelf als framework waarmee ambitieuze webapplicaties kunnen worden gemaakt; daarom is het in eerste instantie geschikt voor ontwikkelaars die al ervaring hebben met het ontwerpen van webapplicaties.

JavaScript-frameworks/library's en trage opbouw

De client-side rendering die single-page-applicaties doen heeft een vertraagde rendering en het leveren van meer JavaScript naar de client als gevolg. Dit is een beperking die alle JavaScript-frameworks en -library's hebben, hoewel het idealiter wordt verzacht door snellere interactie nadat de pagina is geladen.

Omdat dit soort webapplicaties niet direct als HTML dienen, is er een vertraging nadat de eerste HTML-pagina is geladen, voordat JavaScript de toepassing weergeeft. Vooral op mobiele apparaten kan dit een lange vertraging zijn, omdat het framework vaak een grote library heeft om te downloaden, in aanvulling op de applicatiecode die de ontwikkelaar zelf schrijft.

Frameworks werken steeds vaker om dit probleem heen door het maken van zogeheten *server-side rendering engines* (SSR) in Node.js, die de applicatie vooraf als vaste HTML maken, zodat de browser meteen HTML ontvangt, die bij het laden van de pagina direct kan worden getoond. De React-, Ember- en Angular-gebruikersgroepen ontwikkelen allemaal versies van deze technologie.

Nuttige tools voor veel toepassingen

Frameworks en bibliotheken kunnen het werken met JavaScript aanzienlijk vereenvoudigen. De hierboven besproken uitbreidingen vormen slechts een klein deel van de beschikbare tools. Een goede bron voor andere JavaScript-frameworks, -library's en -plug-ins is naast GitHub (github.com/trending/javascript) ook de website www.javascripting.com. Hier kunt u geschikte JavaScript-uitbreidingen vinden voor diverse doeleinden.

U vervolgt uw reis en belandt bij de “HOE...dan?”-vraag.

Hoezo React en hoe gaat u React zich eigen maken?

Elke dag worden er nieuwe programmeertalen geboren. Dart, Go, Kotlin, Elixir en nog veel meer! Het verslaan van de leercurve kan moeilijk zijn, maar er zijn diverse manieren die u helpen om er zonder kleerscheuren doorheen te komen.

Hoezo... React?

Ik laat u enkele van de belangrijkste eigenschappen zien die React te bieden heeft, waardoor het zich volgens mij onderscheidt van de andere library's en frameworks.

- **Declaratief** React creëert zeer interactieve en dynamische UI's voor websites en mobiele applicaties. Creëer eenvoudige weergaven voor elke toestand (*state*) in uw applicatie en React zal op een efficiënte manier de juiste delen bijwerken en weergeven wanneer uw gegevens veranderen. Declaratieve weergaven waarin u beschrijft wát u wilt, maken uw code leesbaarder en gemakkelijker te debuggen.
- **Virtual DOM** In React is er voor elke webpagina (DOM-object) een overeenkomstig 'virtueel DOM-object'. Zo'n DOM-object is een representatie van een DOM-object, het maakt een virtuele kopie van het originele DOM-object. Uw gegevens worden maar één richting op gestuurd – *one-way data binding* – waardoor het bewerken van het virtual DOM snel gaat in plaats van het bijwerken van het originele DOM, omdat er niets op het scherm wordt weergegeven.
- **Event handling** React creëert een eigen eventsysteem dat volledig te gebruiken is met het W3C-objectmodel. Alle web-browser-eigen events worden verpakt in het zogeheten *React Synthetic Event*. Het biedt in meerdere browsers toegang tot het browser-eigen eventsysteem. Dat betekent dat u zich geen zorgen hoeft te maken over namen en velden van niet ondersteunde events. Daarnaast is Reacts eventsysteem gerealiseerd door middel van *event delegation* en heeft het ook een *pool* van eventobjecten om de geheugenlast te verminderen.
- **JSX** JSX kan het best worden gezien als een opmaaksyntax die het meest lijkt op HTML. Het is min of meer een combinatie van JavaScript en XML. JSX maakt het schrijven van React-componenten, de bouwstenen van de React-UI, eenvoudiger door de syntaxis. Ontwikkelaars gebruiken deze JSX-strings,

die bijna identiek zijn aan de HTML die ze in de webpagina zullen injecteren. JSX is een van de beste React-mogelijkheden. Nogal wat applicatieontwikkelaars zijn lui, daarom is dit voor de meesten een goede keuze.

- **Prestaties** React gebruikt one-way data binding met een applicatiearchitectuur genaamd Flux. React helpt u om de weergaven voor de gebruiker bij te werken en kan, met Flux, de workflow van de applicatie controleren. De introductie van virtual DOM biedt extra voordelen door de nieuwe gegevens te vergelijken met het originele DOM en de weergaven automatisch bij te werken.
- **React Native** React Native is een aangepaste weergave-module (*renderer*) voor React, net als React DOM op het web. React Native gebruikt systeemeigen componenten (of bouwstenen) in plaats van webcomponenten, zoals React. Om te beginnen met React Native moet u de basisconcepten van React Native kennen, zoals JSX, Components, State en Props. Als u React kent, moet u nog steeds dingen leren die specifiek zijn voor React Native, zoals de systeemeigen componenten. React Native geeft ook toegang tot de mogelijkheden die deze platformen bieden, naast het omzetten van React-code naar code voor iOS en Android.
- **Component-gebaseerd** In React is alles een component van de webpagina, verdeeld in kleine componenten om een view (of UI's) te maken. Elk deel van de schermen van een applicatie wordt verpakt in een op zichzelf staande module die bekend staat als een component. Omdat componentlogica is geschreven in JavaScript in plaats van templates, kunt u eenvoudig uitgebreide data door uw applicatie sturen zonder inmenging van DOM. *Components*. In React gebruiken we JavaScript om de visuals en interacties in applicaties te definiëren.

Dogfooding – Facebook zelf aan het roer, voor meer rendement uit eigen voer!

Ik hecht waarde aan stabiliteit. Bij Facebook zijn er meer dan 50.000 componenten die gebruikmaken van React. Facebook heeft ook een enorme niet-React-codebase. De website maakt gebruik van een mix van een server-side componentsysteem genaamd XHP, interne UI-library's die nog vóór React in gebruik zijn genomen, en React zelf. Veel andere bedrijven, waaronder Twitter en Airbnb, zijn ook intensieve gebruikers van React. Dit is de reden waarom ze meestal terughoudend zijn om publieke API's of gedrag te veranderen.



Twitter

Twitter is een Amerikaanse online nieuws- en sociale netwerkservice waarop gebruikers berichten plaatsen en communiceren met berichten die bekend staan als 'tweets', opgericht in 2006 door Jack Dorsey, Noah Glass, Biz Stone en Evan Williams.

Denk niet dat het hier gaat om stabiliteit in de zin van 'niets verandert', maar meer in de zin van 'het wordt veel gebruikt in de productie, en als er iets verandert, is er een duidelijk (en bij voorkeur geautomatiseerd) migratiepad'.



Afbeelding 1.2 Dogfooding.

Intensief intern gebruik en de gedrevenheid door de behoeften van de producten bij Facebook geven mij in ieder geval het vertrouwen dat React niet snel zal verdwijnen. React werd gecreëerd bij Facebook om problemen op te lossen. Het brengt tastbare *business value* voor het bedrijf en wordt gebruikt in veel van zijn producten. Dogfooding betekent dat hun visie scherp blijft en dat ze een doelgerichte aanpak hebben om vooruit te gaan.

React maakt gebruik van nieuwe technologie die nog niet eerder is gebruikt bij de ontwikkeling van webapplicaties. Het is een perfect hulpmiddel om een hoogwaardige presentatielaag voor uw applicatie te maken. Het hebben van een goed presterende library als React is de droom van veel bedrijven die zich bezighouden met de ontwikkeling van webapplicaties. Dan moet React de beste oplossing zijn... toch?

Hoe gaat u React zich eigen maken?

Er is geen manier om iets op een eenvoudige manier te leren. U moet geduld hebben. Dat geldt ook voor ontwikkelaars. Persoonlijk vind ik het daarom leuk om ontwikkelaar te zijn, omdat er altijd iets nieuws te leren valt. Als u nieuwsgierig blijft, zult u niet teleurgesteld worden tijdens uw ontwikkeling. Leren is altijd een uitdaging, een uitdaging om uw vaardigheden aan te scherpen, als de uitdaging en de vaardigheden die er zijn in balans zijn. Daarom kost het ook tijd om React te leren.

React heeft geen steile leercurve als u één eenvoudige regel volgt: leer React – alleen React – stap voor stap, en laat u niet afleiden door andere technologieën op deze reis. JavaScript en React zijn voortdurend in ontwikkeling, omdat ze beide innovatief blijven en ook in de toekomst van betekenis willen blijven; dus leer en volg ze eerst op de voet voordat u er iets anders bij leert. Ik heb succesverhalen gehoord van ontwikkelaars, van het niet kennen van React tot het aangeboden krijgen van een React-baan bij een

bedrijf. In de praktijk kregen ze de kans om meer te weten te komen over alle andere fancy technologieën die bij React horen.

Verandering in uw werk- en denkwijze

Om te kunnen inspelen op veranderingen in React en op de continu veranderende wensen van uw klant, moet u eerst in staat worden gesteld innovatief en wendbaar (*agile*) te werken en te denken. Het begint bij het aanvaarden van een verandering in uw werk- en denkwijze.

Beginnen met een nieuwe technologie kan behoorlijk lastig zijn. U belandt meestal in een zee van of overvloed aan tutorials en artikelen, gevolgd door miljoenen persoonlijke meningen. En iedereen zegt dat ze de “juiste en perfecte manier” hebben gevonden. Dit laat u meestal met de vraag zitten of de door u gekozen tutorial een verspilling van tijd zal zijn of niet.

Als u begint met het leren van React, moet u eerst in React denken. Voordat u in de oceaan duikt, moet u de onderliggende concepten van een technologie begrijpen. Dan moet u een op technologie gebaseerde mindset ontwikkelen. Dit gaat u in de volgende hoofdstukken aanpakken.

Samengevat

U heeft een indruk gekregen van React. Zoals gezegd zijn er andere library's en frameworks beschikbaar. Met de wetenschap dat de keuze voor React afhankelijk is van vele factoren, verwacht ik dat u na het lezen van dit boek in staat zal zijn om die keuze te maken.