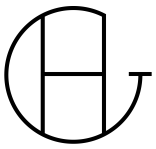

HoGent

Faculteit Bedrijf en Organisatie

Redux uitbreiding gebaseerd op Redux-store

Thibault Gobert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Harm De Weirdt

Instelling: —

Academiejaar: 2017-2018

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Redux uitbreiding gebaseerd op Redux-store

Thibault Gobert

Scriptie voorgedragen tot het bekomen van de graad van
professionele bachelor in de toegepaste informatica

Promotor:
Harm De Weirdt

Instelling: —

Academiejaar: 2017-2018

Tweede examenperiode

Woord vooraf

Samenvatting

In dit onderzoek worden twee React-Redux projecten opgesteld. React maakt gebruik van JavaScript voor het bouwen van user interfaces, waar Redux een concept is voor data storage en communicatie binnen de applicatie. Redux functioneert als een state container voor JavaScript apps. Een store is een concept van Redux die de hele state tree van de applicatie vasthoudt.

Het probleem situeert zich het samenvoegen van twee React-Redux projecten. Deze hebben beide een store die de overeenkomstige state tree bevat. Er kan dus niet rechtstreeks een component geïmporteerd worden van het ene project naar het andere, omdat het daar geen bekende state zal hebben en er zo geen acties op kunnen uitgevoerd worden.

De bekende oplossing voor een soortgelijk probleem is het kopiëren van de reducers van het tweede project en deze plaatsen in het eerste project. [TODO-REFERENTIE HIER] Deze oplossing heeft echter een groot nadeel: als de reducers van het open source project gewijzigd worden of als er reducers worden toegevoegd/verwijderd, dan zullen deze ook opnieuw gekopieerd moeten worden in het eerste project.

Om het probleem te reproduceren worden dus twee React-Redux projecten opgesteld. Het eerste project is het hoofdproject met een basis inlog functionaliteit. Het tweede project is een fork van het bestaande open source project van Scratch. Om beide projecten met elkaar te laten communiceren zal het tweede project gepubliceerd worden op npm onder een eigen registry. Dit zorgt ervoor dat het project geïnstalleerd kan worden als dependency in het eerste project. Op deze manier kan een component snel geïmporteerd worden.

De bedoeling van dit onderzoek is om een manier te vinden waarbij geen extra aanpassingen moeten gebeuren aan het eerste project, maar enkel een geupdatete versie van het tweede project moet gepublished worden op npm. Deze manier zal verkregen worden door een

extensie te schrijven op Redux die de mogelijheid biedt om de functionaliteit van het tweede project te gebruiken in het eerste project.

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Inleiding

0.1 Context

Zoals eerder vermeld, wordt gebruik gemaakt van twee React-Redux projecten. React is een JavaScript library voor het bouwen van user interfaces. React werkt met components. In de component schrijf je de gewenste code die moet gerenderd worden. Een component ontvangt parameters, genaamd *props* en retourneert hiërarchische views die getoond worden door de render methode. Elke component kan onafhankelijk functioneren, wat het dus mogelijk maakt om verschillende components in een andere component in te laden. (React, 2018)

Redux is een state container voor JavaScript apps. Redux kan dus gebruikt worden samen met React of andere view libraries. In Redux zijn er een aantal basis begrippen die moeten uitgelegd worden voor de verdere voortgang van dit onderzoek, zoals actions, reducers, store en containers. Deze begrippen vormen eigenlijk de omkadering van Redux. (Redux, 2018)

De Redux architectuur heeft een unidirectionele data flow, dit wil zeggen de data steeds dezelfde richting/flow aanneemt. Dit zorgt er voor dat de logica van de applicatie voorspelbaar wordt.

Een container component is een component die verantwoordelijk is voor het verkrijgen van data. Een container component gaat subscriben op de store, om zo een deel van de Redux state tree te kunnen lezen. Het gaat eigenlijk de nodige delen van de data nemen en doorgeven als *props*. Een container component is ook verantwoordelijk voor het dispatchen van actions die veranderingen maken aan de state van de applicatie.

Deze actions zijn payloads van informatie die data verzenden van de applicatie naar de store. Actions zijn tevens de enige soort van informatie voor de store. Het zijn JavaScript objecten die een *type* property moeten hebben om aan de duiden welke actie uitgevoerd wordt. Deze types zijn string constanten die in een aparte module worden opgeslaan.

Als er dan gekeken wordt naar de data lifecycle dan worden er eerst actions gedispatched naar de store. De store zal dan de corresponderende reducer aanroepen met twee argumenten, namelijk de huidige state en de action. Hierna zal de root reducer de output van meerdere reducers combineren in een single state tree. Daarna gaat de Redux store de volledige state tree die geretourneerd werd door de root reducer gaan opslaan. Hier kan de container dan data uit lezen en doorgeven aan een presentationele component.

Reducers specificeren hoe de applicatie zijn state verandert ten gevolge van de actions die verzonden zijn naar de store. Actions beschrijven alleen maar het feit dat er iets gebeurd is, ze beschrijven niet hoe de applicatie zijn state verandert. Vooraleer er kan gezegd worden hoe een reducer dit doet, moet worden uitgelegd wat een pure functie is. Een pure functie is een functie die met dezelfde input altijd dezelfde output produceert. (Elliott, 2016)

Een reducer is een pure functie die de vorige state en een actie neemt en daaruit de nieuwe state retourneert. Het is belangrijk dat de reducer puur blijft, een aantal zaken zijn een no-go zoals API-calls en niet-pure functies aanroepen. De uitkomst moet voorspelbaar blijven. Redux roept de reducer aan met een *undefined* state voor de eerste keer. Daar moet de initial state van de applicatie worden ingesteld. (Redux, 2018)

Actions representeren het feit dat er iets gebeurd is en reducers updaten de state aan de hand van deze actions. De store is een object die deze zaken samenbrengt. Deze store heeft een aantal verantwoordelijkheden:

- vasthouden van de state van de applicatie
- toegang geven tot de state van de applicatie
- toelaten om de state te updaten
- registreren van listeners

Het laatste punt laat toe om een callback te registreren die de redux store zal aanroepen elke keer een actie wordt gedispatched. Op deze manier kan de UI van de applicatie upgedate worden naargelang de state van de applicatie.

0.2 Probleemstelling

Het probleem ligt in het gebruiken van components uit een tweede React-Redux project. Door het importeren van een container component in het eerste project worden de actions die gedispatched worden door die component niet herkent in de store. Deze zitten namelijk in de store van het tweede project. Door de reducers te combineren van beide projecten in een grote root reducer wordt dit probleem opgelost. Dit is zo omdat de store dan de corresponderende reducer kan aanroepen naargelang de action die gedispatched wordt. Het probleem hierbij is dat de reducers van het tweede project gekopieerd moeten worden in

de root reducer van het eerste project. Dit resulteert in onstabiele code wanneer er reducers toegevoegd en/of verwijderd worden in het tweede project. Dit onderzoek heeft een meerwaarde voor bedrijven/personen die een React-Redux project hebben en functionaliteit (componenten) willen gebruiken uit een bestaande React-Redux repository.

0.3 Onderzoeksvraag

Hoe kunnen twee root reducers met elkaar gecombineerd worden?

0.4 Onderzoeksdoelstelling

Het beoogde resultaat van de bachelorproef is om een minimaal aantal lines of code te hebben. Idealiter is dit een aangepaste versie van de functie *combineReducers* waarbij er twee root reducers worden gecombineerd met elkaar.

0.5 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 1 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 2 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 3, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvraag. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.

1. Stand van zaken

1.1 React

Volgens (React, 2018) is React een JavaScript library om user interfaces te bouwen. React is component-based. Een component implementeert een *render* methode die data als input neemt en een view retourneert. Om dit te doen wordt JSX gebruikt. Er kunnen simpele views gemaakt worden voor elke state van de applicatie en React zal deze updaten en de juiste componenten renderen wanneer de data verandert.

1.1.1 JSX

JSX is een XML/HTML-achtige syntax gebruikt door React die ECMAScript uitbreidt zodat XML/HTML-achtige text kan bestaan met JavaScript/React code. Deze syntax is bedoeld om gebruikt te worden door preprocessoren (zoals Babel) om HTML text gevonden in JavaScript files om te zetten in standaard JavaScript objecten. Dit wil dus zeggen dat je door JSX te gebruiken HTML structuren en JavaScript code kan schrijven in dezelfde file, Babel zal dan alle uitdrukkingen vertalen naar JavaScript code. Waar vroeger dus JavaScript code in HTML werd geplaatst, laat JSX het toe om HTML in JavaScript te plaatsen. (TODO REF JSX)

1.1.2 Components

Een enkele view van een user interface is opgedeeld in een aantal stukken, in een aantal components. De start component bevat een tree, die tree kan opgedeeld worden in een aantal sub-componenten. Deze kunnen dan weer opgedeeld worden in nog meer sub-

componenten. Dit kan dus resulteren in een complexe tree met verschillende React componenten. (TODO REF JSX ZELFDE)

Simple component

React components implementeren de *render* methode, ze retourneren wat er moet afgebeeld worden. Daarvoor wordt het eerder aangekaarte JSX gebruikt. Input data die doorgegeven is aan de component kan opgeroepen worden door de *props* aan te spreken van deze component. Sidenote: JSX is optioneel, het gewenste resultaat kan ook bereikt worden door JavaScript code alleen. JSX maakt het wel overzichtelijker om de props aan te spreken. (React, 2018)

Stateful component

In toevoeging met data als input nemen (via *props*), kan een component zijn interne state data aanspreken. Wanneer de state data van een component verandert, wordt de render methode opnieuw aangeropen zodat de juiste data getoond wordt. (React, 2018)

Dit hoofdstuk bevat je literatuurstudie. De inhoud gaat verder op de inleiding, maar zal het onderwerp van de bachelorproef *diepgaand* uitspitten. De bedoeling is dat de lezer na lezing van dit hoofdstuk helemaal op de hoogte is van de huidige stand van zaken (state-of-the-art) in het onderzoeksdomein. Iemand die niet vertrouwd is met het onderwerp, weet er nu voldoende om de rest van het verhaal te kunnen volgen, zonder dat die er nog andere informatie moet over opzoeken (Pollefliet, 2011).

Je verwijst bij elke bewering die je doet, vakterm die je introduceert, enz. naar je bronnen. In LaTeX kan dat met het commando `\textcite{}` of `\autocite{}`. Als argument van het commando geef je de "sleutel" van een "record" in een bibliografische databank in het BibTeX-formaat (een tekstbestand). Als je expliciet naar de auteur verwijst in de zin, gebruik je `\textcite{}`. Soms wil je de auteur niet expliciet vernoemen, dan gebruik je `\autocite{}`. In de volgende paragraaf een voorbeeld van elk.

Knuth (1998) schreef een van de standaardwerken over sorteer- en zoekalgoritmen. Experten zijn het erover eens dat cloud computing een interessante opportuniteit vormen, zowel voor gebruikers als voor dienstverleners op vlak van informatietechnologie (Creeger, 2009).

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet

mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

2. Methodologie

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Maecenas non massa. Vestibulum pharetra nulla at lorem. Duis quis quam id lacus dapibus interdum. Nulla lorem. Donec ut ante quis dolor bibendum condimentum. Etiam egestas

tortor vitae lacus. Praesent cursus. Mauris bibendum pede at elit. Morbi et felis a lectus interdum facilisis. Sed suscipit gravida turpis. Nulla at lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent nonummy luctus nibh. Proin turpis nunc, congue eu, egestas ut, fringilla at, tellus. In hac habitasse platea dictumst.

Vivamus eu tellus sed tellus consequat suscipit. Nam orci orci, malesuada id, gravida nec, ultricies vitae, erat. Donec risus turpis, luctus sit amet, interdum quis, porta sed, ipsum. Suspendisse condimentum, tortor at egestas posuere, neque metus tempor orci, et tincidunt urna nunc a purus. Sed facilisis blandit tellus. Nunc risus sem, suscipit nec, eleifend quis, cursus quis, libero. Curabitur et dolor. Sed vitae sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla facilisi.

3. Conclusie

Curabitur nunc magna, posuere eget, venenatis eu, vehicula ac, velit. Aenean ornare, massa a accumsan pulvinar, quam lorem laoreet purus, eu sodales magna risus molestie lorem. Nunc erat velit, hendrerit quis, malesuada ut, aliquam vitae, wisi. Sed posuere. Suspendisse ipsum arcu, scelerisque nec, aliquam eu, molestie tincidunt, justo. Phasellus iaculis. Sed posuere lorem non ipsum. Pellentesque dapibus. Suspendisse quam libero, laoreet a, tincidunt eget, consequat at, est. Nullam ut lectus non enim consequat facilisis. Mauris leo. Quisque pede ligula, auctor vel, pellentesque vel, posuere id, turpis. Cras ipsum sem, cursus et, facilisis ut, tempus euismod, quam. Suspendisse tristique dolor eu orci. Mauris mattis. Aenean semper. Vivamus tortor magna, facilisis id, varius mattis, hendrerit in, justo. Integer purus.

Vivamus adipiscing. Curabitur imperdiet tempus turpis. Vivamus sapien dolor, congue venenatis, euismod eget, porta rhoncus, magna. Proin condimentum pretium enim. Fusce fringilla, libero et venenatis facilisis, eros enim cursus arcu, vitae facilisis odio augue vitae orci. Aliquam varius nibh ut odio. Sed condimentum condimentum nunc. Pellentesque eget massa. Pellentesque quis mauris. Donec ut ligula ac pede pulvinar lobortis. Pellentesque euismod. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent elit. Ut laoreet ornare est. Phasellus gravida vulputate nulla. Donec sit amet arcu ut sem tempor malesuada. Praesent hendrerit augue in urna. Proin enim ante, ornare vel, consequat ut, blandit in, justo. Donec felis elit, dignissim sed, sagittis ut, ullamcorper a, nulla. Aenean pharetra vulputate odio.

Quisque enim. Proin velit neque, tristique eu, eleifend eget, vestibulum nec, lacus. Vivamus odio. Duis odio urna, vehicula in, elementum aliquam, aliquet laoreet, tellus. Sed velit. Sed vel mi ac elit aliquet interdum. Etiam sapien neque, convallis et, aliquet vel, auctor non, arcu. Aliquam suscipit aliquam lectus. Proin tincidunt magna sed wisi. Integer blandit

lacus ut lorem. Sed luctus justo sed enim.

Morbi malesuada hendrerit dui. Nunc mauris leo, dapibus sit amet, vestibulum et, commodo id, est. Pellentesque purus. Pellentesque tristique, nunc ac pulvinar adipiscing, justo eros consequat lectus, sit amet posuere lectus neque vel augue. Cras consectetuer libero ac eros. Ut eget massa. Fusce sit amet enim eleifend sem dictum auctor. In eget risus luctus wisi convallis pulvinar. Vivamus sapien risus, tempor in, viverra in, aliquet pellentesque, eros. Aliquam euismod libero a sem.

Nunc velit augue, scelerisque dignissim, lobortis et, aliquam in, risus. In eu eros. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur vulputate elit viverra augue. Mauris fringilla, tortor sit amet malesuada mollis, sapien mi dapibus odio, ac imperdiet ligula enim eget nisl. Quisque vitae pede a pede aliquet suscipit. Phasellus tellus pede, viverra vestibulum, gravida id, laoreet in, justo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer commodo luctus lectus. Mauris justo. Duis varius eros. Sed quam. Cras lacus eros, rutrum eget, varius quis, convallis iaculis, velit. Mauris imperdiet, metus at tristique venenatis, purus neque pellentesque mauris, a ultrices elit lacus nec tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent malesuada. Nam lacus lectus, auctor sit amet, malesuada vel, elementum eget, metus. Duis neque pede, facilisis eget, egestas elementum, nonummy id, neque.

A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

A.1 Introductie

Vooraleer er dieper op het probleem kan worden ingegaan is er meer context nodig rond een aantal basisbegrippen van React en Redux. Volgens de website React (2018) is React een JavaScript library. Met behulp van React worden views gecreëerd voor elke state in onze applicatie. Volgens de website Redux (2018) is Redux een state container voor JavaScript apps. Redux helpt om de afgebeelde data te beheren en beschrijft hoe er gereageerd wordt op user actions. Een store is een object die de volledige state-tree van de applicatie bevat. De enige manier om een state te veranderen in de store is door een action te dispatchen. Een action zal er eigenlijk voor zorgen dat er data van de applicatie verzonden wordt naar de store. Dit is tevens de enige vorm van input voor de store. Actions beschrijven dus het feit dat er iets gebeurd is, maar ze beschrijven niet hoe de state van de applicatie verandert. Dit is de taak van een reducer, deze toont aan hoe de state veranderd is ten gevolge van een action die verzonden werd naar de store. Een container is dan weer verantwoordelijk voor het verkrijgen van data. Om die data te verkrijgen wordt er gebruik gemaakt van de connect functie (voor de store) en een functie die de data uit de state neemt en deze mapt naar zijn eigen props. Een container is ook verantwoordelijk voor het dispatchen van actions. (Redux, 2018) Het probleem zit bij het samenvoegen van 2 react-redux apps tot 1 app. Er is geen manier bekend om beide stores te behouden in 1 project. Dit is relevant omdat er toch een aantal hits op Overflow (2018) te vinden zijn voor dit probleem. 2 React projecten die elk afzonderlijk een Redux store gebruiken kunnen

als ze samengevoegd worden momenteel maar de volledige functionaliteit van 1 store gebruiken. Onderzoeksvraag:

- Op welke manier kunnen de stores van 2 react-redux apps onafhankelijk functioneren in 1 app?

A.2 State-of-the-art

Zoals eerder vermeld is er zeer weinig documentatie te vinden over dit probleem. Er zijn wel een aantal vragen in die richting op Stack Overflow gesteld. De antwoorden die op deze vragen gegeven zijn, bieden geen oplossing op lange termijn of bieden geen schaalbaarheid/uitbreidbaarheid. Er zal veel tijd gestoken worden in het verdiepen van de GitHub repository van Redux, geschreven door Abramov, 2016. Er is wel documentatie te vinden over de Redux-store sharen in meerdere components binnen hetzelfde project. Hier staat een uitgewerkt voorbeeld van op Stack Overflow door b.g (2017). Het wordt echter onduidelijk wanneer we de Redux-store willen sharen tussen meerdere projecten.

A.3 Methodologie

Er zal geprobeerd worden een uitbreiding te schrijven op de bestaande Redux library. Zoals eerder vermeld houdt een redux store eigenlijk de hele state-tree van de applicatie vast. De bedoeling van deze uitbreiding zal zijn om een draaiende app te hebben met 2 onafhankelijke stores van 2 projecten. Deze realisatie zal gebeuren door het ene project om te zetten naar een npm-package. Op deze manier kunnen geëxporteerde componenten makkelijk geïmporteerd worden in het andere project. Met de uitbreiding zal geprobeerd worden om een store toe te voegen aan het project, zonder dat de andere store daar enige invloed van ondervindt. Om dit te kunnen realiseren zal het react-redux patroon goed moeten bestudeerd worden. Daarbij kan ook naar Flux (2015) en MobX (2017) gekeken worden om eventuele analogieën door te trekken. In dit onderzoek zullen een aantal grafieken worden gemaakt met de gemiddelde rendertijd van verschillende oplossingen (met bv: nesting en reducers importeren). Anderszijds wordt er ook naar de lines of code gekeken. Een groot criteria hierbij is dan het dupliceren van code of het vervuilen van de applicatie door code toe te voegen die ze eigenlijk niet nodig zou moeten hebben.

A.4 Verwachte resultaten

Zoals eerder aangegeven worden er een aantal oplossingen verwacht. Enerzijds kan een nesting van de stores een mogelijke oplossing bieden, maar de vraag is dan of alle functionaliteit wordt behouden. Door het schrijven van de uitbreiding kan de rendertijd hoger zijn dan normaal. Een andere oplossing is door de reducers van het ene project samen te voegen met de reducers van het andere project, waarbij een hoger aantal lines of code verwacht wordt.

A.5 Verwachte conclusies

Een verwachte conclusie bij nesting is dat mogelijks niet alle functionaliteit behouden wordt. Een kopie nemen van de reducers van het ene project is geen efficiënte oplossing en wel om deze reden: stel dat er een update gepushed wordt naar de npm-package dan zou men alle reducers opnieuw moeten kopiëren naar het andere project. Tevens vervuilt dit ook de hoofdapplicatie.

Bibliografie

Abramov, D. (2016). Redux [Github]. Verkregen van https://github.com/reactjs/redux

b.g, S. (2017). How to share redux store in multiple components.

Creeger, M. (2009). CTO Roundtable: Cloud Computing. *Communications of the ACM*, *52*(8), 50–56.

Elliott, E. (2016, maart 26). Master the JavaScript interview: What is a pure function? Verkregen van https://medium.com/javascript-scene/master-the-javascript-interview-what-is-a-pure-function-d1c076bec976

Flux. (2015). Flux. Verkregen van https://facebook.github.io/flux/

Knuth, D. E. (1998). *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.

MobX. (2017). MobX.js. Verkregen van https://mobx.js.org/

Overflow, S. (2018). Stack Overflow. Verkregen van https://stackoverflow.com

Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Gent: Academia Press.

React. (2018). React.js. Verkregen van https://reactjs.org/

Redux. (2018). Redux.js. Verkregen van https://redux.js.org/