

Fachrichtung Informatik**Schuljahr 2020**

React

by Facebook

Ausgeführt von:

Matthias Wetzlmayer, 4BHIF

Manuel Seifriedsberger, 4BHIF

Elias Bauer, 4BHIF

Julian Moshhammer, 4BHIF

Daniel Riffert, 4BHIF

At Home, am 27.04.2020

Abgabevermerk:

Datum: 30.04.2020

Inhaltsverzeichnis

1	Übersicht	3
1.1	Vorteile	3
1.2	Nachteile.....	4
2	Framework aufsetzen	4
2.1	Vorbereitung.....	4
2.2	Arbeiten mit React / Übersicht	5
2.2.1	Index.html	5
2.2.2	Index.js.....	5
2.2.3	[Komponente].js.....	5
3	REST-Calls	5
4	Besonderheiten des Frameworks	6
5	Probleme und Lösungen.....	6
6	Empfehlungen	6
7	Cooler Features.....	7
8	Geschichte des Frameworks	7
9	Community	8
10	Bekannte Applikationen.....	8
11	Tooling.....	8
11.1	IDEs 8	
11.2	Library-Infrastruktur.....	8
12	Geschwindigkeit	9
13	Lernkurve	9
14	Konzepte von React.....	10
15	Persönliche Meinung.....	10
16	Würdet Ihr den Einsatz des Frameworks empfehlen?.....	10
17	Quellen	11

1 Übersicht

React		Angular
Category	Library	Framework
Language	Javascript, JSX	TypeScript
DOM	Virtual DOM	DOM, Regular

1.1 Vorteile

Vorteile	
	Virtual DOM (=Neurendern nur von geänderten Komponenten)
	Komponenten können in einem einzigen File zusammengesetzt werden und sind wiederverwendbar.
	Andere Librarys müssen genutzt werden, was aber kein Nachteil ist, da man an keine Library gebunden ist und sich die aussuchen kann, die man bevorzugt.
	Kann schnell eine große Menge an Komponenten anzeigen
	Dank Chrome Extension ist das Debuggen leichter
	SEO freundlich (Search Engine Optimization)
	Front-end JavaScript wird sehr vereinfacht

1.2 Nachteile

Nachteile	
	Nicht für jede Application sinnvoll, nur bei dynamischem Content.
	Man muss Daten manuell verändern
	Kann mehr Code beanspruchen
	View orientiert
	Sehr umfangreich und schnell wachsend, viele Librarys → Eher schlechte Dokumentation, unübersichtlich

2 Framework aufsetzen

2.1 Vorbereitung

Node.js muss als erstes installiert werden, da npm benötigt wird, um das Erstellen von React-Projekten zu vereinfachen. Sinnvoll ist auch die React Developer Tools-Erweiterung für Google Chrome, welche das Arbeiten mit React im Browser erheblich erleichtert. Da nun npm installiert ist - welches ein Package-Manager ist – kann man das Terminal öffnen, sich in den entsprechenden Ordner bewegen und mit dem Befehl `npx create-react-app [app-name]` ganz einfach ein React-Projekt erstellen. Mit `cd [app-name]` nun einfach in das entsprechende Projekt navigieren und mit `npm start` den Server starten. Es kann sein, dass man noch `npm-install` ausführen muss, damit alle benötigten Pakete installiert werden.

Node.js – <https://nodejs.org/en>

React Developer Tools - <https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

2.2 Arbeiten mit React / Übersicht

2.2.1 Index.html

In diesem File befindet sich ein `<div>`-Element mit der id „root“. In dieses Div wird mithilfe des `index.js`-File der gesamte Inhalt geladen.

2.2.2 Index.js

Dieses File lädt den gesamten Inhalt in das `<div>`-Element mit der entsprechenden id. Hier wird `ReactDOM.render([param1],[param2])` aufgerufen. Der erste Parameter ist eine Komponente, der zweite referenziert das `<div>`-Element mithilfe der id, in das der Output geladen werden sollte.

2.2.3 [Komponente].js

Die Default-Komponente, welche generiert wird ist `App.js`. Komponenten sind im Prinzip nichts anderes als Klassen. Jede Klasse besitzt ein State-Objekt, die man anderen Komponenten übergeben kann. Jede Klasse braucht außerdem eine `render()`-Methode, die Daten bereitstellt, welche zurückgegeben werden.

3 REST-Calls

Wir haben alle REST-Calls in eine eigene Klasse gepackt (Stichwort Architektur).

Um einen REST-Call abzusetzen verwenden wir `Axios`. Es gibt auch andere Möglichkeiten, doch uns gefiel dieses Package sehr gut. Dazu muss man **`npm install axios`** in der Konsole ausführen. In der Klasse muss man `axios` dann noch importieren **`import axios from 'axios';`**

Die REST-Calls sind dann ganz einfach:

- `axios.get(link)`
- `axios.post(link, DTO)`
- `axios.put(link, DTO)`
- `axios.delete(link)`

Dieser Call gibt ein Promise zurück, welches wir ganz einfach mit `.then(res => {...});` abhandeln können. Um mögliche Exceptions des BackEnds abfangen zu können, kann man genauso wie `.then()` ein `.catch(exception => {...})` machen.

4 Besonderheiten des Frameworks

- Im Normalfall keine Trennung von UI und Logik
- Es können HTML-artige Konstrukte direkt in den JavaScript-Code implementiert werden, welcher mittels eines Compilers in echtes JavaScript übersetzt wird
- UI wird deklarativ beschrieben → Komponenten beschreiben deklarativ, wie der Zustand der Anwendung auf das UI gemappt wird -> UI als Funktion
- „virtuelles DOM“ wird von React in den „realen DOM“ überführt

5 Probleme und Lösungen

Man muss bei den Imports aufpassen, denn es gibt 2 Arten: Einmal bekommt man den „default export“, z. B. **import React from 'react';** oder man importiert etwas bestimmtes, z. B.: **import { view } from '@risingstack/react-easy-state';**

Das Problem dabei ist, wenn etwas keinen expliziten Import hat und man trotzdem die Klammern macht, wirft das Fehler. Eine nachfolgende Stackoverflow-Seite hat uns sehr geholfen, um den Fehler auszubessern und zu verstehen.

Ansonsten hatten wir keine größeren Probleme mit React selber.

6 Empfehlungen

- Benutzt React-Easy-State
 - Das State-Management kann normalerweise sehr kompliziert werden, denn man muss Methoden mitgeben, um einen zentralen State zu bearbeiten. Mithilfe des React-Easy-States kann man eine eigene State-Klasse machen, wo man alle Daten hält. Die View-Komponenten, die auf ein entsprechendes Feld zugreifen, werden bei Änderung neu gerendert.
- Installiert die Chrome Extension “React Developer Tool”
- Teilt die Aufgaben in viele einzelne Komponenten auf, das Zusammenbauen bereitet keine Schwierigkeiten und der Code wird viel übersichtlicher

7 Coole Features

- Error Boundaries
 - Wenn ein Fehler beim Rendern auftritt, kann man diesen mit einer Komponente, die sich in der Hierarchie weiter oben befindet, auffangen und darauf reagieren (Stichwort: componentDidCatch)
- Man kann beeinflussen, ob eine Komponente neu gerendert wird (ShouldComponentUpdate) oder mit einem setState() – Call, wo dass Value auf „Null“ gesetzt wird.
- Open-Source
- Änderungen in der UI ohne die Seite neu laden zu müssen
- JSX (= JavaScript XML)
- Lifecycle Methods
 - Damit kann man einfach Code zu bestimmten Lebenszyklen der Komponenten ausführen (componentDidMount, shouldComponentUpdate, ...)
- Conditional Statements
 - Man kann z.B. im Code if – Statements einbauen. So kann man z.B. den Background abhängig von einer Variable setzen

8 Geschichte des Frameworks

React wurde entwickelt, um das Problem eines komplettem und performance-schädigendem Re-Rendering zu beheben.

Jahr	
2011	Von Facebook entwickelt
	Für Facebook Newsfeed eingesetzt
2012	Für Instagram verwendet
2013	React wird Open-Source-Projekt
2014	BSD-Lizenz
2017	MIT-Lizenz

MIT und BSD Lizenzen sind Open-Source Lizenzen

9 Community

Hinter React steht eine Community von mehreren Millionen Entwicklern. Es werden auch auf der React-Website Tutorials zur Verfügung gestellt. Es gibt viele Extensions / Librarys, die von der Community erstellt und verwaltet werden.

10 Bekannte Applikationen



Netflix



Twitter



Instagram



Facebook



Uber



Reddit



11 Tooling

11.1 IDEs

- Visual Studio Code
 - Flexible IDE und für viele Sprachen zu verwenden
- JetBrains Webstorm
 - Speziell für Javascript

11.2 Library-Infrastruktur

Es gibt zahlreiche Librarys, die ganz einfach über npm installiert werden können. Diese erleichtern nicht nur die Entwicklung in React, sondern verschönern auch noch die UI (Alerts,...).

12 Geschwindigkeit

React legt sehr viel Wert auf die Geschwindigkeit. So wird z.B. ein virtueller DOM verwendet, wodurch nur die Komponenten neu gerendert werden, die auch neu gerendert werden müssen. Da die Seiten vom Client neu gerendert werden, dauert zwar das Laden der Seite zum 1. Mal etwas länger, danach ist das Render aber sehr schnell. So können Interaktionen sehr schnell abgearbeitet werden.

13 Lernkurve

- Matthias Wetzlmayer
 - Vom Grundkonzept war es nicht schwer zu erlernen (JS im HTML, Komponenten). Schwieriger war es, einen guten State-Manager zu finden, allerdings war dabei das Erlernen auch kein großes Problem.
- Daniel Riffert
 - Grundsätzlich war es kein großer Aufwand, React zu verstehen. Wichtig, und vielleicht etwas schwieriger ist das State-Management, doch auch dies ist nicht allzu schwer zu bewältigen gewesen.
- Manuel Seifriedsberger
 - Bei mir war ein Tutorial genug, ich habe mir zwar auch ein 2. Angesehen, zum Thema React-easy-state, jedoch war dies unverständlich. Zum Glück konnte mir Matthias dies gut erklären.
- Julian Moshhammer
 - Tutorials auf unserer allerliebsten bekanntesten Plattform Youtube hatten uns alle sehr geholfen. Jedoch halfen wir uns auch untereinander. Mit viel Teamgeist und miteinander ist uns denke ich dadurch ein gutes Projekt gelungen. Mein Dank hierbei gilt vor allem meinen Mitarbeitern / Mitschülern. Danke Jungs!
- Elias Baran Bauer
 - Dank den guten Tutorials, die wir fanden, habe ich mich nachdem ich noch ein paar Fragen gestellt habe gut genug ausgekannt, um meine Aufgaben ohne größere Probleme zu vollenden.
- Unterlagen:
 - <https://www.youtube.com/watch?v=sBws8MSXN7A>
 - <https://www.youtube.com/watch?v=jleTmmDhxX8>
 - <https://material-ui.com/components/alert/#api>

14 Konzepte von React

React folgt der Philosophie von Unix: Write programs that do one thing and do it well. Write programs to work together. – Doug Mcllory. React fokussiert sich auf Bau von User Interfaces.

Durch den virtual-Dom werden Operationen, die den DOM beeinflussen so selten wie möglich gemacht. So wird z.B. ein Tree, der geändert wurde, mit der alten Version verglichen und nur die Änderungen werden auch übernommen / neu gerendert.

15 Persönliche Meinung

- Matthias Wetzlmayer
 - Voi geil. React ist ein einfaches Framework, mit dem sich einfach moderne Web-Frontends entwickeln lassen.
- Daniel Riffert
 - Richtig geil. Die Seite lässt sich bearbeiten ohne neu laden zu müssen. Mit dem React-Easy-State lässt sich das State-Management leicht und übersichtlich umsetzen.
- Manuel Seifriedsberger
 - Mir persönlich hat es unglaublich gut gefallen. Ich habe mich dabei vielleicht auch ein bisschen zu sehr reingehängt. Es hat mir, verglichen zu anderen Fächern / Gelehrtem, mit Abstand am meisten Spaß gemacht.
- Julian Moshhammer
 - *hat React mit 5 / 5 Sternen bewertet*
 - Richtig cooles Teil dieses React. Taugt mir. Ein bisschen React-Easy-State hier und State-Manager da und schwubdiwub eine React-Applikation ist da.
- Elias Baran Bauer
 - Man findet sich leicht rein und es ist leicht damit umzugehen. Mir gefällt es sehr gut.

16 Würdet Ihr den Einsatz des Frameworks empfehlen?

Auf jedem Fall. Mit Librarys lässt sich das Leben unheimlich erleichtern und viele zusätzliche Funktionen einbinden.

17 Quellen

- <https://justintimecoder.com/5-awesome-features-of-react-16/>
- [https://en.wikipedia.org/wiki/React_\(web_framework\)#Basic_usage](https://en.wikipedia.org/wiki/React_(web_framework)#Basic_usage)
- <https://medium.com/edge-coders/all-the-fundamental-react-js-concepts-jammed-into-this-single-medium-article-c83f9b53eac2>
- <https://blog.logrocket.com/angular-vs-react-vs-vue-a-performance-comparison/>
- <https://www.freecodecamp.org/news/what-exactly-is-client-side-rendering-and-how-it-different-from-server-side-rendering-bd5c786b340d/>
- <https://stackoverflow.com/questions/53521063/named-import-in-react>
- <https://blog.logrocket.com/angular-vs-react-vs-vue-a-performance-comparison/>