
Inhaltsverzeichnis

Teil I Einstieg

| | | |
|----------|--|-----------|
| 1 | Einleitung | 3 |
| 1.1 | Was ist React? | 3 |
| 1.2 | Warum React? | 7 |
| 1.3 | Beziehung zu anderen Technologien | 8 |
| 1.4 | Vergleich mit anderen Webtechnologien | 14 |
| 1.5 | Wie man dieses Buch benutzt | 16 |
| 1.6 | Voraussetzungen für dieses Buch | 18 |
| 1.7 | Änderungen gegenüber der ersten Auflage | 18 |
| 1.8 | Website zum Buch | 18 |
| 1.9 | Danksagungen | 19 |
| 2 | Schnelldurchgang – React im Überblick | 21 |
| 2.1 | Zusammengefasst: Unterschiede zwischen Hooks- und Klassen-API .. | 26 |
| 2.2 | Ein React-Projekt beginnen mit Create React App | 27 |
| 2.3 | Zusammenfassung | 32 |
| 3 | Die Beispielanwendung: »Vote as a Service« | 33 |
| 3.1 | Die Beispielanwendung installieren und ausführen | 33 |
| 3.2 | Fachliches Modell | 37 |
| 3.3 | Die Anwendung schrittweise entwickeln | 38 |
| 3.4 | Zusammenfassung | 40 |

Teil II React

| | | |
|----------|--|------------|
| 4 | Eine React-Komponente | 43 |
| 4.1 | Hands-on: eine Komponente | 43 |
| 4.2 | React-Komponenten in der Übersicht | 54 |
| 4.3 | Hooks | 54 |
| 4.4 | Zustand verwalten mit useState | 58 |
| 4.5 | Ereignisse und Event Handler | 62 |
| 4.6 | JSX zur Beschreibung der UI | 63 |
| 4.7 | Rückgabewerte von Komponenten | 69 |
| 4.8 | Einhängen der Anwendung in den DOM | 72 |
| 4.9 | Arbeiten mit CSS | 75 |
| 4.10 | Zusammenfassung | 80 |
| 5 | Arbeiten mit Komponentenhierarchien | 81 |
| 5.1 | Hands-on: Hierarchien von Komponenten | 81 |
| 5.2 | Kommunikation zwischen Komponenten | 92 |
| 5.3 | Das »Render Properties«-Pattern | 94 |
| 5.4 | Performance-Optimierung: Caching von Werten mit useMemo | 96 |
| 5.5 | Performance-Optimierung: Rendern vom Komponenten unterdrücken | 98 |
| 5.6 | Performance-Optimierung: Code-Splitting mit React.lazy und Suspense | 106 |
| 5.7 | Der React Strict Mode | 109 |
| 5.8 | Zusammenfassung | 110 |
| 6 | Formulare mit React | 111 |
| 6.1 | Hands-on: ein Editor für Umfragen | 111 |
| 6.2 | Hintergrund: Databinding | 125 |
| 6.3 | Controlled Components | 126 |
| 6.4 | Uncontrolled Components | 128 |
| 6.5 | Auf native DOM-Elemente zugreifen: das ref-Property | 130 |
| 6.6 | Komplexen Zustand mit useReducer verwalten | 131 |
| 6.7 | Zusammenfassung | 135 |

| | | |
|----------|---|------------|
| 7 | Arbeiten mit Seiteneffekten: asynchrone Serverzugriffe | 137 |
| 7.1 | Hands-on: Serveranbindung | 138 |
| 7.2 | Seiteneffekte mit useEffect | 150 |
| 7.3 | Code wiederverwenden mit Custom Hooks | 155 |
| 7.4 | Server-Requests mit useState und useReducer | 158 |
| 7.5 | Ausblick: Daten laden mit Suspense | 162 |
| 7.6 | Zusammenfassung | 162 |

Teil III Über React hinaus

| | | |
|-----------|---|------------|
| 8 | React-Anwendungen testen | 165 |
| 8.1 | Hands-on: Testen mit Jest und React Testing Library | 165 |
| 8.2 | Überblick: React-Anwendungen testen | 176 |
| 8.3 | React-Test-Bibliotheken | 178 |
| 8.4 | Snapshot Testing | 182 |
| 8.5 | Zusammenfassung | 184 |
| 9 | Der React Router | 185 |
| 9.1 | Hands-on: der React Router im Schnelldurchgang | 185 |
| 9.2 | Anpassungen an der Vote-Anwendung | 189 |
| 9.3 | Links und Routen | 190 |
| 9.4 | Navigation über die History | 197 |
| 9.5 | Authentifizierung | 198 |
| 9.6 | Die React Context API | 203 |
| 9.7 | Lazy Loading mit dem React Router | 209 |
| 9.8 | Testen | 211 |
| 9.9 | Zusammenfassung | 214 |
| 10 | Externes State-Management mit Redux | 215 |
| 10.1 | Motivation | 215 |
| 10.2 | Hands-on: eine Redux-Anwendung | 219 |
| 10.3 | Redux in der Vote-Anwendung | 230 |
| 10.4 | Arbeiten mit dem globalen Zustand | 231 |
| 10.5 | Asynchrone Actions | 241 |
| 10.6 | Alternative zu Hooks: die connect-Funktion | 249 |
| 10.7 | Integration von Redux und React Router | 252 |

| | | |
|-----------|---|------------|
| 10.8 | Testen von Redux-Anwendungen | 252 |
| 10.9 | Exkurs: Codestruktur von großen React-Anwendungen | 255 |
| 10.10 | Alternative zu Redux: MobX | 259 |
| 10.11 | Zusammenfassung | 261 |
| 11 | React-Anwendungen mit TypeScript | 263 |
| 11.1 | Hands-on: eine React-Anwendung mit TypeScript | 263 |
| 11.2 | TypeScript in React-Anwendungen | 269 |
| 11.3 | React-Komponenten mit TypeScript | 271 |
| 11.4 | Der React Router mit TypeScript | 280 |
| 11.5 | Redux | 281 |
| 11.6 | Zusammenfassung | 286 |
| 12 | GraphQL mit dem Apollo Client für React | 287 |
| 12.1 | Hands-on: ein GraphQL-Client | 287 |
| 12.2 | GraphQL in der Vote-Anwendung | 293 |
| 12.3 | Mutations | 295 |
| 12.4 | Der Apollo Client Cache | 297 |
| 12.5 | GraphQL für externes Statemanagement? | 303 |
| 12.6 | Der Apollo React Client mit TypeScript | 304 |
| 12.7 | Zusammenfassung | 307 |

Anhang

| | | |
|----------|---|------------|
| A | Serverseitiges Rendern mit React | 311 |
| A.1 | Ein Beispiel | 311 |
| A.2 | Gründe für serverseitiges Rendern | 316 |
| A.3 | Serverseitiges Rendern im Überblick | 318 |
| A.4 | Herausforderungen | 319 |
| A.5 | Asynchrones Datenladen mit Redux und dem React Router | 321 |
| B | Komponenten als Klassen | 327 |
| B.1 | React-Komponenten als ES6-Klasse | 329 |
| B.2 | Methoden-Binding für Event Handler | 335 |
| B.3 | Zugriff auf native DOM-Elemente | 336 |
| B.4 | Arbeiten mit Seiteneffekten | 337 |
| B.5 | Beispiel: Error Boundaries | 340 |
| B.6 | Klassenkomponenten mit TypeScript | 341 |

| | | |
|----------|--|------------|
| C | Einführung in ES.Next | 345 |
| C.1 | Einleitung | 345 |
| C.2 | Block-Scope | 345 |
| C.3 | Template Strings | 346 |
| C.4 | Destructuring | 347 |
| C.5 | Klassen | 348 |
| C.6 | Vererbung | 349 |
| C.7 | Erweiterte Objekt-Literale | 350 |
| C.8 | Module, Exporte und Importe | 350 |
| C.9 | Arrow-Funktionen | 352 |
| C.10 | Default- und Rest Parameter | 354 |
| C.11 | Spread-Operator für Arrays | 354 |
| C.12 | Object.assign und Spread-Operator bei Objekten | 355 |
| C.13 | Promises | 356 |
| C.14 | async/await | 358 |
| C.15 | Generatorfunktionen | 359 |
| C.16 | Die fetch-API | 360 |
| D | Einführung in TypeScript | 363 |
| D.1 | Motivation | 363 |
| D.2 | Die Sprache TypeScript | 365 |
| D.3 | Grundlagen des TypeScript-Typsysteams | 366 |
| D.4 | Externe Typbeschreibungen | 376 |
| E | Übersicht GraphQL | 379 |
| E.1 | Abfragen | 380 |
| E.2 | Das Schema: Beschreibung der API | 384 |
| | Index | 387 |