

Inhalt

Materialien zum Buch	18
Geleitwort des Fachgutachters	19
Vorwort	21

1 Die ersten Schritte mit React

25

1.1 Was ist React?	25
1.1.1 Single-Page-Applikationen	26
1.1.2 Die Geschichte von React	27
1.2 Warum React?	32
1.2.1 Der Releasezyklus	32
1.3 Die wichtigsten Begriffe und Konzepte der React-Welt	33
1.3.1 Komponenten und Elemente	34
1.3.2 Der Datenfluss	36
1.3.3 Der Renderer	37
1.3.4 Der Reconciler	38
1.4 Ein Blick in das React-Universum	40
1.4.1 Das State-Management	40
1.4.2 Der Router	41
1.4.3 Material UI	41
1.4.4 Jest	41
1.5 Thinking in React	41
1.5.1 Die Oberfläche in eine Komponentenhierarchie zerlegen	42
1.5.2 Eine statische Version in React implementieren	42
1.5.3 Den minimalen UI State bestimmen	42
1.5.4 Den Speicherort des States bestimmen	43
1.5.5 Den inversen Datenfluss modellieren	43
1.6 Codebeispiele	43
1.7 Zusammenfassung	44

2	Die ersten Schritte im Entwicklungsprozess	45
2.1	Schnellstart	45
2.1.1	Die Initialisierung	45
2.2	Playgrounds für React	46
2.2.1	CodePen – ein Playground für die Webentwicklung	47
2.3	Lokale Entwicklung	50
2.3.1	React in eine HTML-Seite einbinden	50
2.4	Der Einstieg in die Entwicklung mit React	53
2.4.1	Anforderungen	54
2.4.2	Installation von Create React App	56
2.4.3	Alternativen zu Create React App	63
2.4.4	React Scripts	64
2.4.5	Serverkommunikation im Entwicklungsbetrieb	67
2.4.6	Verschlüsselte Kommunikation während der Entwicklung	68
2.5	Die Struktur der Applikation	69
2.6	Fehlersuche in einer React-Applikation	70
2.6.1	Arbeiten mit den React Developer Tools	72
2.7	Die Applikation bauen	73
2.8	Zusammenfassung	74
3	Die Grundlagen von React	75
3.1	Vorbereitung	75
3.1.1	Die Applikation aufräumen	76
3.2	Einstieg in die Applikation	76
3.2.1	Index.jsx – das Rendering der Applikation	76
3.2.2	App.jsx – die Wurzelkomponente	79
3.3	Funktionskomponenten	80
3.3.1	Eine Komponente pro Datei	83
3.4	JSX – Strukturen in React definieren	88
3.4.1	Ausdrücke in JSX	90
3.4.2	Iterationen – Schleifen in Komponenten	93
3.4.3	Bedingungen in JSX	95

3.5	Props – Informationsfluss in einer Applikation	99
3.5.1	Props und Kindkomponenten	100
3.5.2	Typsicherheit mit PropTypes	102
3.6	Lokaler State	105
3.7	Event-Binding – Reaktion auf Benutzerinteraktionen	107
3.7.1	Auf Events reagieren	108
3.7.2	Arbeiten mit Event-Objekten	113
3.8	Immutability	116
3.8.1	Immer in einer React-Applikation	117
3.9	Zusammenfassung	119

4 Ein Blick hinter die Kulissen – weiterführende Themen

4.1	Der Lebenszyklus einer Komponente	121
4.2	Der Lebenszyklus einer Funktionskomponente mit dem Effect-Hook	122
4.2.1	Mount – das Einhängen einer Komponente	123
4.2.2	Update – Aktualisieren der Komponente	126
4.2.3	Unmount – Aufräumen am Ende des Lebenszyklus	130
4.3	Serverkommunikation	133
4.3.1	Serverimplementierung	133
4.3.2	Serverkommunikation mit der Fetch API	136
4.3.3	Wissenswertes zur Serverkommunikation	139
4.3.4	Serverkommunikation mit Axios	143
4.4	Container Components	145
4.4.1	Auslagern von Logik in eine Container Component	146
4.4.2	Einbindung der Container Component	148
4.4.3	Implementierung der Presentational Component	149
4.5	Higher-Order Components	151
4.5.1	Eine einfache Higher-Order Component	151
4.5.2	Einbindung einer Higher-Order Component in die BooksList-Komponente	153
4.5.3	Einbindung der Higher-Order Component	155
4.6	Render Props	155
4.6.1	Alternative Namen für Render Props	157
4.6.2	Integration der Render Props in die Applikation	159

- 4.7 Kontext** 160
 - 4.7.1 Die Context-API 161
 - 4.7.2 Einsatz der Context-API in der Beispielapplikation 164
- 4.8 Fragments** 168
- 4.9 Zusammenfassung** 170
- 5 Klassenkomponenten** 171

- 5.1 Klassenkomponenten in React** 171
- 5.2 Grundlegende Struktur einer Klassenkomponente** 172
- 5.3 Props in einer Klassenkomponente** 173
 - 5.3.1 Prop-Strukturen mit PropTypes definieren 174
 - 5.3.2 Standardwerte für Props 175
- 5.4 State – der Zustand der Klassenkomponente** 176
 - 5.4.1 Den State über die state-Eigenschaft der Klasse initialisieren 177
 - 5.4.2 Den State im Konstruktor initialisieren 178
- 5.5 Der Komponenten-Lifecycle** 179
 - 5.5.1 Constructor 182
 - 5.5.2 »getDerivedStateFromProps« 183
 - 5.5.3 »render« 184
 - 5.5.4 »componentDidMount« 184
 - 5.5.5 »shouldComponentUpdate« 186
 - 5.5.6 »getSnapshotBeforeUpdate« 186
 - 5.5.7 »componentDidUpdate« 187
 - 5.5.8 »componentWillUnmount« 189
 - 5.5.9 Unsafe Hooks 190
- 5.6 Error Boundaries** 191
 - 5.6.1 Mit »componentDidCatch« Fehler loggen 192
 - 5.6.2 Alternative Darstellung im Fehlerfall mit
»getDerivedStateFromError« 194
- 5.7 Die Context-API in einer Klassenkomponente verwenden** 196
- 5.8 Unterschiede zwischen Funktions- und Klassenkomponenten** 198
 - 5.8.1 State 198
 - 5.8.2 Lifecycle 198
- 5.9 Zusammenfassung** 198

6	Die Hooks-API von React	201
<hr/>		
6.1	Ein erster Überblick	202
6.1.1	Die drei Basis-Hooks	202
6.1.2	Weitere Bestandteile der Hooks-API	203
6.2	»useReducer« – der Reducer Hook	204
6.2.1	Die Reducer-Funktion	207
6.2.2	Actions und Dispatching	207
6.2.3	Asynchronität im Reducer-Hook	207
6.3	»useCallback« – Memoisieren von Funktionen	212
6.4	»useMemo« – Memoisieren von Objekten	213
6.5	»useRef« – Referenzen und immutable Values	215
6.5.1	Formularhandling mit dem Ref-Hook	215
6.5.2	Werte mit dem Ref-Hook zwischenspeichern	216
6.6	»useImperativeHandle« – Steuerung von ForwardRefs	218
6.6.1	ForwardRefs	218
6.6.2	Der ImperativeHandle-Hook	219
6.7	»useLayoutEffect« – die synchrone Alternative zu useEffect	221
6.8	»useDebugValue« – Debugging-Informationen in den React Developer Tools	222
6.9	»useDeferredValue« – Updates nach Priorität durchführen	223
6.10	»useTransition« – die Priorität von Operationen heruntersetzen	226
6.11	»useId« – Eindeutige Identifier erzeugen	228
6.12	Bibliotheks-Hooks	229
6.12.1	»useSyncExternalStore«	229
6.12.2	»useInsertionEffect«	230
6.13	Custom Hooks	230
6.13.1	Ein Beispiel für einen Custom Hook	231
6.14	Rules of Hooks – was Sie beachten sollten	232
6.14.1	Regel #1: Hooks nur auf oberster Ebene ausführen	233
6.14.2	Regel #2: Hooks dürfen nur in Funktionskomponenten oder Custom Hooks verwendet werden	233
6.15	Umstieg auf Hooks	234
6.16	Zusammenfassung	235

7

Typsicherheit in React-Applikationen
mit TypeScript

237

7.1	Was bringt ein Typsystem?	237
7.2	Die verschiedenen Typsysteme	238
7.3	Typsicherheit in einer React-Applikation mit Flow	239
7.3.1	Einbindung in eine React-Applikation	239
7.3.2	Die wichtigsten Features von Flow	242
7.3.3	Flow in React-Komponenten	243
7.4	TypeScript in einer React-Applikation einsetzen	244
7.4.1	TypeScript in eine React-Applikation einbinden	245
7.4.2	Konfiguration von TypeScript	248
7.4.3	Die wichtigsten Features von TypeScript	249
7.4.4	Typdefinitionen – Informationen über Drittanbieter-Software	249
7.5	TypeScript und React	250
7.5.1	TypeScript zu einer bestehenden Applikation hinzufügen	250
7.5.2	Basisfeatures	251
7.5.3	Funktionskomponenten	255
7.5.4	Context	259
7.5.5	Klassenskomponenten	261
7.6	Zusammenfassung	263

8

Styling von React-Komponenten

265

8.1	CSS-Import	265
8.1.1	Die Vor- und Nachteile des CSS-Imports	266
8.1.2	Umgang mit Klassennamen	268
8.1.3	Verbesserte Behandlung von Klassennamen mit der »classnames«-Bibliothek	271
8.1.4	Verwendung von Sass als CSS-Präprozessor	272
8.2	Inline-Styling	274
8.3	CSS-Module	277
8.4	CSS in JavaScript mit Emotion	279
8.4.1	Emotion installieren	280
8.4.2	Arbeiten mit der »css«-Prop	280
8.4.3	Der styled-Ansatz von Emotion	282

8.4.4	Pseudoselektoren in Styled Components	285
8.4.5	Dynamisches Styling	285
8.4.6	Weitere Features von Styled Components	287
8.5	Tailwind	288
8.5.1	Tailwind installieren und einbinden	289
8.6	Zusammenfassung	291

9 Eine React-Applikation durch Tests absichern 293

9.1	Die ersten Schritte mit Jest	295
9.1.1	Installation und Ausführung	295
9.1.2	Organisation der Tests	296
9.1.3	Jest – die Grundlagen	297
9.1.4	Aufbau eines Tests – Triple A	298
9.1.5	Die Matcher von Jest	300
9.1.6	Gruppierung von Tests – Testsuites	301
9.1.7	Setup- und Teardown-Routinen	302
9.1.8	Tests überspringen und exklusiv ausführen	304
9.1.9	Umgang mit Exceptions	306
9.1.10	Testen von asynchronen Operationen	307
9.2	Testen von Hilfsfunktionen	311
9.3	Snapshot-Testing	313
9.3.1	Snapshot-Tests für Komponenten	313
9.4	Komponenten testen	318
9.4.1	Test der »BooksListItem«-Komponente	319
9.4.2	Interaktion testen	321
9.5	Umgang mit Serverabhängigkeiten	323
9.5.1	Fehler bei der Kommunikation simulieren	326
9.6	Zusammenfassung	328

10 Formulare in React 331

10.1	Uncontrolled Components	331
10.1.1	Der Umgang mit Referenzen in React	332
10.2	Controlled Components	345
10.2.1	Synthetic Events	356

10.3	Der Upload von Dateien	356
10.4	Formularvalidierung mit React Hook Form	363
10.4.1	Formularvalidierung mit React Hook Form	366
10.4.2	Formularvalidierung mit einem Schema	368
10.4.3	Styling des Formulars	371
10.4.4	Die Formularvalidierung automatisiert testen	373
10.5	Zusammenfassung	374

11 Komponentenbibliotheken in einer React-Applikation 377

11.1	Installation und Integration von Material-UI	377
11.2	Listendarstellung mit der »Table«-Komponente	379
11.2.1	Die Liste in der Tabelle filtern	383
11.2.2	Die Tabelle sortieren	385
11.3	Grids und Breakpoints	389
11.4	Icons	392
11.5	Datensätze löschen	395
11.5.1	Eine Löschoperation vorbereiten	395
11.5.2	Einen Bestätigungsdialog implementieren	396
11.5.3	Datensätze löschen	398
11.6	Neue Datensätze erzeugen	401
11.6.1	Das Erzeugen von Datensätzen vorbereiten	401
11.6.2	Implementierung der »Form«-Komponente	402
11.6.3	Integration des Formulardialogs	406
11.7	Datensätze editieren	408
11.8	Zusammenfassung	413

12 Navigation innerhalb einer Applikation – der Router 415

12.1	Installation und Einbindung	416
12.1.1	Die Routerkomponenten	416

12.2	Navigation in der Applikation	417
12.2.1	Es wird immer die beste Route aktiviert	419
12.2.2	Eine Navigationsleiste für die Applikation	420
12.2.3	Integration der Navigationsleiste	422
12.3	»Not found«	423
12.4	Testen des Routings	425
12.5	Bedingte Umleitungen	430
12.6	Dynamische Routen	433
12.6.1	Subrouten definieren	433
12.7	Zusammenfassung	440

13 Eigene React-Bibliotheken erzeugen 441

13.1	Eine eigene Komponentenbibliothek erzeugen	441
13.1.1	Initialisierung der Bibliothek	442
13.1.2	Die Struktur der Bibliothek	445
13.1.3	Hooks in der Bibliothek	448
13.1.4	Das Bauen der Bibliothek	449
13.2	Einbinden der Bibliothek	451
13.2.1	Reguläre Installation des Pakets	453
13.3	Testen der Bibliothek	454
13.3.1	Die Testumgebung vorbereiten	454
13.3.2	Unittest für die Bibliothekskomponente	455
13.3.3	Unittest des Custom Hooks der Bibliothek	456
13.4	Storybook	458
13.4.1	Installation und Konfiguration von Storybook	458
13.4.2	Button-Story in Storybook	459
13.5	Zusammenfassung	461

14 Zentrales State-Management mit Redux 463

14.1	Die Flux-Architektur	464
14.1.1	Der zentrale Datenspeicher – der Store	464
14.1.2	Die Anzeige der Daten mit den Views	465

14.1.3	Actions – die Beschreibung von Änderungen	465
14.1.4	Der Dispatcher – die Schnittstelle zwischen Actions und dem Store	466
14.2	Installation von Redux	467
14.2.1	Die Struktur der Applikation	468
14.3	Den zentralen Store konfigurieren	468
14.3.1	Debugging mit den Redux Dev Tools	470
14.4	Der Umgang mit Änderungen am Store mit Reducern	472
14.4.1	Der »Books«-Slice	472
14.4.2	Einbindung des »BooksSlice«	474
14.5	Komponenten und den Store verknüpfen	476
14.5.1	Anzeige der Daten aus dem Store	476
14.5.2	Selektoren	478
14.5.3	Selektoren mit Reselect umsetzen	480
14.6	Änderungen mit Actions beschreiben	482
14.6.1	Löschen von Datensätzen	482
14.7	Datensätze erstellen und bearbeiten	486
14.8	Zusammenfassung	492

15 Umgang mit Asynchronität und Seiteneffekten in Redux 495

15.1	Middleware in Redux	495
15.1.1	Eine eigene Middleware implementieren	496
15.2	Redux mit Redux Thunk	497
15.2.1	Manuelle Integration von Redux Thunk	498
15.2.2	Daten vom Server lesen	498
15.2.3	Datensätze löschen	506
15.2.4	Datensätze anlegen und modifizieren	511
15.3	Generators – Redux Saga	515
15.3.1	Installation und Einbindung von Redux Saga	517
15.3.2	Daten vom Server laden	518
15.3.3	Bestehende Daten löschen	524
15.3.4	Datensätze erstellen und modifizieren mit Redux Saga	530
15.4	State-Management mit RxJS – Redux Observable	535
15.4.1	Redux Observable installieren und einbinden	536

15.4.2	Lesender Zugriff auf den Server mit Redux Observable	537
15.4.3	Löschen mit Redux Observable	539
15.4.4	Datensätze anlegen und editieren mit Redux Observable	540
15.5	JWT zur Authentifizierung	542
15.5.1	Implementierung des »login«-Moduls	542
15.6	Zusammenfassung	551

16 Serverkommunikation mit GraphQL und dem Apollo-Client 553

16.1	Einführung in GraphQL	553
16.1.1	Die Charakteristik von GraphQL	553
16.1.2	Die Nachteile von GraphQL	554
16.1.3	Die Prinzipien von GraphQL	555
16.2	Apollo, ein GraphQL-Client für React	559
16.2.1	Installation und Einbindung in die Applikation	560
16.2.2	Lesender Zugriff auf den GraphQL-Server	561
16.2.3	Zustände einer Anfrage	563
16.2.4	Typunterstützung im Apollo-Client	565
16.2.5	Löschen von Datensätzen	568
16.3	Die Apollo Client Devtools	571
16.4	Lokales State-Management mit Apollo	573
16.4.1	Den lokalen State initialisieren	573
16.4.2	Den lokalen State benutzen	574
16.5	Authentifizierung	578
16.6	Zusammenfassung	580

17 Internationalisierung 583

17.1	Einsatz von react-i18next	584
17.1.1	Sprachdateien vom Backend laden	588
17.1.2	Die Sprache des Browsers verwenden	590
17.1.3	Die Navigation um eine Sprachumschaltung erweitern	591
17.2	Platzhalter verwenden	593

17.3	Werte formatieren	596
17.3.1	Zahlen und Währungen formatieren	596
17.3.2	Datumswerte formatieren	598
17.4	Singular und Plural	600
17.5	Zusammenfassung	603
 18 Universal React Apps mit Server-Side Rendering		605
<hr/>		
18.1	Wie funktioniert Server-Side Rendering?	606
18.2	Umsetzung von Server-Side Rendering	607
18.2.1	Die Server-Applikation initialisieren und konfigurieren	608
18.2.2	Die clientseitige Applikation implementieren	614
18.2.3	Dynamik im Server-Side Rendering	617
18.3	Server-Side Rendering mit Next.js	620
18.3.1	Eine Next.js-Applikation initialisieren	621
18.3.2	Die Page-Komponente implementieren	621
18.3.3	Die Serverseite implementieren	623
18.3.4	API-Routen in Next.js	625
18.4	Zusammenfassung	626
 19 Performance		629
<hr/>		
19.1	Der Callback-Hook	630
19.2	Pure Components	632
19.3	React.memo	634
19.4	»React.lazy« – Suspense for Code Splitting	637
19.4.1	Lazyloading in einer Applikation	637
19.4.2	Lazyloading mit dem React Router	641
19.5	Suspense for Data Fetching	645
19.5.1	React Query installieren und verwenden	645
19.5.2	React Query und Suspense	650
19.5.3	Concurrency Patterns	652
19.6	Virtuelle Tabellen	653
19.7	Zusammenfassung	658

20 Progressive Web Apps 659

20.1	Merkmale einer Progressive Web App	659
20.2	Initialisieren der Applikation	660
20.3	Installierbarkeit	661
20.3.1	Die sichere Auslieferung einer Applikation	661
20.3.2	Das Web App Manifest	664
20.3.3	Service Worker in der React-Applikation	667
20.3.4	Installation der Applikation	668
20.3.5	Die Benutzer*innen fragen	670
20.4	Offlinefähigkeit	674
20.4.1	Integration von Workbox	674
20.4.2	Umgang mit dynamischen Daten	679
20.5	Werkzeuge für die Entwicklung	685
20.6	Zusammenfassung	686

21 Native Apps mit React Native 689

21.1	Der Aufbau von React Native	689
21.2	Die Installation von React Native	690
21.2.1	Die Projektstruktur	690
21.2.2	Die Applikation starten	691
21.3	Anzeige einer Übersichtsliste	694
21.3.1	Statische Listenansicht	694
21.3.2	Styling in React Native	697
21.3.3	Suchfeld für die »List«-Komponente	703
21.3.4	Serverkommunikation	705
21.4	Debugging in der simulierten React-Native-Umgebung	707
21.5	Bearbeiten von Datensätzen	710
21.5.1	Implementierung der »Form«-Komponente	711
21.6	Publizieren	717
21.6.1	Build der App	718
21.7	Zusammenfassung	718

Index	721
-------	-----