

React

Das umfassende Handbuch

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Inhalt

Materialien zum Buch	19
Geleitwort des Fachgutachters	21
Vorwort	23
1 Die ersten Schritte mit React	27
1.1 Was ist React?	27
1.1.1 Single-Page-Applikationen	28
1.1.2 Die Geschichte von React	29
1.2 Warum React?	35
1.2.1 Der Release-Zyklus	36
1.3 Die wichtigsten Begriffe und Konzepte der React-Welt	37
1.3.1 Komponenten und Elemente	37
1.3.2 Der Datenfluss	40
1.3.3 Der Renderer	42
1.3.4 Der Reconciler	42
1.4 Ein Blick in das React-Universum	44
1.4.1 Das State-Management	45
1.4.2 Der Router	45
1.4.3 Material UI	45
1.4.4 Vitest	45
1.5 Thinking in React	46
1.5.1 Die Oberfläche in eine Komponentenhierarchie zerlegen	46
1.5.2 Eine statische Version in React implementieren	46
1.5.3 Den minimalen UI State bestimmen	47
1.5.4 Den Speicherort des States bestimmen	47
1.5.5 Den inversen Datenfluss modellieren	47
1.6 Codebeispiele	47
1.7 Zusammenfassung	48
2 Die ersten Schritte im Entwicklungsprozess	49
2.1 Schnellstart	49
2.1.1 Die Initialisierung	50
2.2 Playgrounds für React	51
2.2.1 CodePen – ein Playground für die Webentwicklung	51

- 2.3 **Lokale Entwicklung** 54
 - 2.3.1 React in eine HTML-Seite einbinden 54
- 2.4 **Der Einstieg in die Entwicklung mit React** 58
 - 2.4.1 Technische Voraussetzungen 59
 - 2.4.2 Das Build-Werkzeug Vite 60
 - 2.4.3 Alternativen zu Vite 67
 - 2.4.4 Vite Scripts 68
 - 2.4.5 Serverkommunikation im Entwicklungsbetrieb 71
- 2.5 **Die Struktur der Applikation** 73
- 2.6 **Fehlersuche in einer React-Applikation** 74
 - 2.6.1 Arbeiten mit den React Developer Tools 76
- 2.7 **Die Applikation bauen** 77
- 2.8 **Zusammenfassung** 78

- 3 Die Grundlagen von React** 79
 - 3.1 **Vorbereitung** 79
 - 3.2 **Einstieg in die Applikation** 80
 - 3.2.1 »main.jsx« – das Rendering der Applikation 80
 - 3.2.2 App.jsx – die Wurzelkomponente 83
 - 3.3 **Funktionskomponenten** 84
 - 3.3.1 Eine Komponente pro Datei 87
 - 3.4 **JSX – Strukturen in React definieren** 92
 - 3.4.1 Ausdrücke in JSX 95
 - 3.4.2 Iterationen – Schleifen in Komponenten 98
 - 3.4.3 Bedingungen in Komponenten 100
 - 3.5 **Props – Informationsfluss in einer Applikation** 103
 - 3.5.1 Props und Kindkomponenten 104
 - 3.5.2 Typsicherheit von Props zur Laufzeit 106
 - 3.6 **Lokaler State** 109
 - 3.7 **Event-Binding – Reaktion auf Benutzerinteraktionen** 111
 - 3.7.1 Auf Events reagieren 111
 - 3.7.2 Arbeiten mit Event-Objekten 117
 - 3.8 **Immutability** 121
 - 3.9 **Zusammenfassung** 125

4	Typsicherheit in React-Applikationen mit TypeScript	127
4.1	Was bringt ein Typsystem?	127
4.2	Die verschiedenen Typsysteme	128
4.3	TypeScript in einer React-Applikation einsetzen	129
4.3.1	TypeScript in eine React-Applikation einbinden	129
4.3.2	Konfiguration von TypeScript	132
4.3.3	Die wichtigsten Features von TypeScript	133
4.3.4	Typdefinitionen – Informationen über Drittanbieter-Software	133
4.4	TypeScript und React	134
4.4.1	Basisfeatures	134
4.4.2	Funktionskomponenten	140
4.5	Zusammenfassung	143
5	Ein Blick hinter die Kulissen – weiterführende Themen	145
5.1	Der Lebenszyklus einer Komponente	145
5.2	Der Lebenszyklus einer Funktionskomponente mit dem Effect-Hook	146
5.2.1	Mount – das Einhängen einer Komponente	146
5.2.2	Update – das Aktualisieren der Komponente	150
5.2.3	Unmount – das Aufräumen am Ende des Lebenszyklus	154
5.3	Serverkommunikation	158
5.3.1	Serverimplementierung	158
5.3.2	Serverkommunikation mit der Fetch-API	160
5.3.3	Umgebungsvariablen bei der Serverkommunikation verwenden	165
5.4	Container Components	167
5.4.1	Auslagern von Logik in eine Container Component	168
5.4.2	Einbindung der Container Component	170
5.4.3	Implementierung der Presentational Component	171
5.5	Higher-Order Components	172
5.5.1	Eine einfache Higher-Order Component	173
5.5.2	Einbindung einer Higher-Order Component in die BooksList-Komponente	176
5.5.3	Einbindung der Higher-Order Component	177

5.6	Render Props	178
5.6.1	Alternative Namen für Render Props	180
5.6.2	Integration der Render Props in die Applikation	182
5.7	Kontext	184
5.7.1	Die Context-API	184
5.7.2	Einsatz der Context-API in der Beispielapplikation	188
5.8	Fragments	192
5.9	Zusammenfassung	194
6	Serverkommunikation mit React	195
6.1	Trennen von Komponente und Kommunikation	197
6.1.1	Umsetzung von API-Funktionen	197
6.1.2	Lesender Serverzugriff	198
6.1.3	Schreibende Serverzugriffe	199
6.2	Bibliotheken für die Serverkommunikation	206
6.3	Validierung der Serverdaten mit Zod	207
6.4	Daten mit TanStack Query vom Server laden	209
6.4.1	Setup von TanStack Query	210
6.4.2	Daten vom Server laden	211
6.4.3	TanStack Query und Suspense for Data Fetching	213
6.4.4	Daten modifizieren	215
6.4.5	Das Caching von TanStack Query	218
6.5	Zusammenfassung	219
7	Formulare in React	221
7.1	Uncontrolled Components	221
7.1.1	Der Umgang mit Referenzen in React	222
7.2	Controlled Components	234
7.2.1	Synthetic Events	240
7.3	Der Upload von Dateien	240
7.4	Formularhandling mit React Hook Form	247
7.4.1	Formularvalidierung mit React Hook Form	251
7.4.2	Formularvalidierung mit einem Schema	254
7.4.3	Styling des Formulars	256
7.5	Zusammenfassung	258

8	Die Hooks-API von React	259
8.1	Ein erster Überblick	260
8.1.1	Die drei Basis-Hooks	260
8.1.2	Weitere Bestandteile der Hooks-API	261
8.2	»useReducer« – der Reducer Hook	263
8.2.1	Die Reducer-Funktion	265
8.2.2	Actions und Dispatching	265
8.2.3	Asynchronität im Reducer-Hook	266
8.3	»useCallback« – Memoisieren von Funktionen	271
8.4	»useMemo« – Memoisieren von Objekten	273
8.5	»useRef« – Referenzen und immutable Values	275
8.5.1	Formularhandling mit dem Ref-Hook	275
8.5.2	Werte mit dem Ref-Hook zwischenspeichern	276
8.6	»useImperativeHandle« – Steuerung von ForwardRefs	277
8.6.1	ForwardRefs	278
8.6.2	Der ImperativeHandle-Hook	280
8.7	»useLayoutEffect« – die synchrone Alternative zu useEffect	281
8.8	»useDebugValue« – Debugging-Informationen in den React Developer Tools	282
8.9	»useDeferredValue« – Updates nach Priorität durchführen	283
8.10	»useTransition« – die Priorität von Operationen heruntersetzen	287
8.11	»useId« – eindeutige Identifier erzeugen	289
8.12	»useEffectEvent« – stabile Callbacks für Effekte und Event-Logik	290
8.13	»useOptimistic« – optimistische Updates ohne komplexes State-Management	292
8.14	Bibliotheks-Hooks	295
8.14.1	»useSyncExternalStore«	295
8.14.2	»useInsertionEffect«	295
8.15	Custom Hooks	296
8.15.1	Ein Beispiel für einen Custom Hook	296
8.16	Rules of Hooks – was Sie beachten sollten	298
8.16.1	Regel #1: Hooks nur auf oberster Ebene ausführen	298
8.16.2	Regel #2: Hooks dürfen nur in Funktionskomponenten oder Custom Hooks verwendet werden	299
8.17	Zusammenfassung	300

9	Styling von React-Komponenten	301
9.1	CSS-Import	301
9.1.1	Die Vor- und Nachteile des CSS-Imports	303
9.1.2	Umgang mit Klassennamen	305
9.1.3	Verbesserte Behandlung von Klassennamen mit der »clsx«-Bibliothek	307
9.1.4	Verwendung von Sass als CSS-Präprozessor	309
9.2	Inline-Styling	311
9.3	CSS-Module	313
9.4	CSS in JavaScript mit Emotion	316
9.4.1	Emotion installieren	316
9.4.2	Arbeiten mit der »css«-Prop	317
9.4.3	Der styled-Ansatz von Emotion	319
9.4.4	Pseudoselektoren in Emotion Styled Components	321
9.4.5	Dynamisches Styling	322
9.4.6	Weitere Features von Emotion	324
9.5	Tailwind	325
9.5.1	Tailwind installieren und einbinden	325
9.6	Zusammenfassung	327
10	Eine React-Applikation durch Tests absichern	329
10.1	Die ersten Schritte mit Vitest	331
10.1.1	Installation und Ausführung	332
10.1.2	Organisation der Tests	334
10.1.3	Vitest – die Grundlagen	334
10.1.4	Aufbau eines Tests – Triple A	336
10.1.5	Die Matcher von Vitest	338
10.1.6	Gruppierung von Tests – Testsuites	339
10.1.7	Setup- und Teardown-Routinen	340
10.1.8	Tests überspringen und exklusiv ausführen	341
10.1.9	Umgang mit Exceptions	343
10.1.10	Testen von asynchronen Operationen	345
10.2	Testen von Hilfsfunktionen	347
10.3	Snapshot-Testing	348
10.3.1	Snapshot-Tests für Komponenten	349
10.4	Komponenten testen	353
10.4.1	Test der »BooksListItem«-Komponente	354

10.4.2	Interaktion testen	357
10.4.3	Formulare testen	358
10.5	Umgang mit Serverabhängigkeiten	361
10.5.1	Fehler bei der Kommunikation simulieren	364
10.6	Ende-zu-Ende-Tests	366
10.6.1	Tests mit dem Vitest Browser Mode	366
10.7	Zusammenfassung	368
11	Komponentenbibliotheken in einer React-Applikation	369
11.1	Installation und Integration von Material-UI	369
11.2	Listendarstellung mit der »Table«-Komponente	371
11.2.1	Die Liste in der Tabelle filtern	373
11.3	Grids und Breakpoints	376
11.4	Icons	378
11.5	Einen Bestätigungsdialog implementieren	381
11.6	Formulare mit Material-UI	386
11.7	shadcn/ui als Alternative zu Material-UI	390
11.7.1	Installation von shadcn/ui	390
11.7.2	shadcn/ui-Komponenten nutzen	391
11.8	Zusammenfassung	392
12	Navigation innerhalb einer Applikation – der Router	393
12.1	Installation und Einbindung	394
12.1.1	Die verschiedenen Router	395
12.2	Navigation in der Applikation	396
12.2.1	Es wird immer die beste Route aktiviert	398
12.2.2	Navigation zwischen verschiedenen Routen	398
12.2.3	Layout-Komponenten im React Router	400
12.3	»Not found«	402
12.4	Testen des Routings	404
12.5	Bedingte Umleitungen	408
12.6	Dynamische Routen	411

- 12.7 Imperative Navigation 413
- 12.8 Routing mit dem TanStack Router 414
 - 12.8.1 Installation und Konfiguration 415
 - 12.8.2 Routen definieren 417
- 12.9 Zusammenfassung 421
- 13 Eigene React-Bibliotheken erzeugen 423**
 - 13.1 Eine eigene Komponentenbibliothek erzeugen 423
 - 13.1.1 Initialisierung der Bibliothek 424
 - 13.1.2 Die Struktur der Bibliothek 428
 - 13.1.3 Hooks in der Bibliothek 429
 - 13.1.4 Das Bauen der Bibliothek 430
 - 13.2 Einbinden der Bibliothek 431
 - 13.2.1 Reguläre Installation des Pakets 433
 - 13.3 Testen der Bibliothek 433
 - 13.3.1 Unit-Test für die Bibliothekskomponente 434
 - 13.3.2 Unit-Test des Custom Hooks der Bibliothek 434
 - 13.4 Storybook 436
 - 13.4.1 Installation und Konfiguration von Storybook 436
 - 13.4.2 Die Button-Story in Storybook 437
 - 13.5 Zusammenfassung 439
- 14 Zentrales State-Management mit Redux 441**
 - 14.1 Die Flux-Architektur 442
 - 14.1.1 Der zentrale Datenspeicher – der Store 442
 - 14.1.2 Die Anzeige der Daten mit den Views 443
 - 14.1.3 Actions – die Beschreibung von Änderungen 443
 - 14.1.4 Der Dispatcher – die Schnittstelle zwischen Actions und dem Store 444
 - 14.2 Installation von Redux 445
 - 14.2.1 Die Struktur der Applikation 446
 - 14.3 Den zentralen Store konfigurieren 446
 - 14.3.1 Debugging mit den Redux Dev Tools 448
 - 14.4 Der Umgang mit Änderungen am Store mit Reducern 449
 - 14.4.1 Der »Books«-Slice 449
 - 14.4.2 Den »BooksSlice« einbinden 452

14.5	Komponenten und den Store verknüpfen	453
14.5.1	Anzeige der Daten aus dem Store	453
14.5.2	Selektoren	455
14.6	Änderungen mit Actions beschreiben	457
14.6.1	Löschen von Datensätzen	458
14.7	Datensätze erstellen und bearbeiten	461
14.8	Zusammenfassung	466
15	Umgang mit Asynchronität und Seiteneffekten in Redux	469
15.1	Middleware in Redux	469
15.1.1	Eine eigene Middleware implementieren	470
15.2	Redux mit Redux Thunk	471
15.2.1	Manuelle Integration von Redux Thunk	472
15.2.2	Daten vom Server lesen	472
15.2.3	Datensätze löschen	480
15.2.4	Datensätze anlegen und modifizieren	484
15.3	Zusammenfassung	488
16	Serverkommunikation mit GraphQL und dem Apollo-Client	491
16.1	Einführung in GraphQL	491
16.1.1	Die Charakteristik von GraphQL	491
16.1.2	Die Nachteile von GraphQL	492
16.1.3	Die Prinzipien von GraphQL	493
16.2	Apollo, ein GraphQL-Client für React	497
16.2.1	Installation und Einbindung in die Applikation	497
16.2.2	Lesender Zugriff auf den GraphQL-Server	499
16.2.3	Zustände einer Anfrage	501
16.2.4	Löschen von Datensätzen	503
16.3	Die Apollo Client Devtools	505
16.4	Lokales State-Management mit Apollo	506
16.4.1	Den lokalen State initialisieren	507
16.4.2	Den lokalen State benutzen	508
16.5	Zusammenfassung	511

17 Internationalisierung	513
17.1 Einsatz von react-i18next	514
17.1.1 Sprachdateien vom Backend laden	518
17.1.2 Die Sprache des Browsers verwenden	519
17.1.3 Die Navigation um eine Sprachumschaltung erweitern	520
17.2 Platzhalter verwenden	522
17.3 Werte formatieren	525
17.3.1 Zahlen und Währungen formatieren	525
17.3.2 Datumswerte formatieren	527
17.4 Singular und Plural	529
17.5 Zusammenfassung	532
18 Performance	533
18.1 Der Callback-Hook	533
18.2 React.memo	537
18.3 Der React Compiler	540
18.3.1 Voraussetzungen für den React Compiler	540
18.3.2 Installation des Compilers	541
18.3.3 Optimierungen mit dem React Compiler	541
18.4 Rules of React	543
18.5 »React.lazy« – Suspense for Code Splitting	544
18.5.1 Lazy Loading in einer Applikation	545
18.5.2 Lazy Loading mit dem React Router	549
18.6 Suspense for Data Fetching	552
18.7 Virtuelle Tabellen	553
18.8 Zusammenfassung	558
19 Authentifizierung in einer React-Applikation	561
19.1 Grundlagen tokenbasierter Authentifizierung	562
19.1.1 JWT – JSON Web Token	562
19.1.2 Arten von Tokens	563
19.1.3 OpenID Connect (OIDC)	563
19.2 Authentifizierungs-State und Token-Handling in React	564
19.2.1 Setup des Identitätsproviders	564
19.2.2 Authentifizierung in der React-Applikation	567

19.3	Geschützte Ressourcen und Requests	569
19.3.1	Implementierung des Backends	569
19.3.2	Kommunikation mit dem Backend	570
19.4	Arbeiten mit Rollen	573
19.5	Zusammenfassung	575
20	Progressive Web Apps	577
20.1	Merkmale einer Progressive Web App	577
20.2	Initialisieren der Applikation	578
20.3	Installierbarkeit	580
20.3.1	Die sichere Auslieferung einer Applikation	580
20.3.2	Das Web-App-Manifest	581
20.3.3	Service-Worker in der React-Applikation	584
20.3.4	Installation der Applikation	584
20.3.5	Die Benutzer fragen	586
20.4	Offlinefähigkeit	590
20.4.1	Eigene Service-Worker	590
20.4.2	Umgang mit dynamischen Daten	594
20.4.3	Offlineunterstützung im Service-Worker	600
20.5	Zusammenfassung	602
21	Native Apps mit React Native	605
21.1	Der Aufbau von React Native	605
21.2	Die Installation von React Native	606
21.2.1	Die Projektstruktur	606
21.2.2	Die Applikation starten	607
21.3	Anzeige einer Übersichtsliste	610
21.3.1	Statische Listenansicht	611
21.3.2	Styling in React Native	614
21.3.3	Suchfeld für die »List«-Komponente	620
21.3.4	Serverkommunikation	622
21.4	Debugging in der simulierten React-Native-Umgebung	624
21.5	Bearbeiten von Datensätzen	625
21.5.1	Implementierung der »Form«-Komponente	626
21.6	Publizieren	633
21.6.1	Build der App	634
21.7	Zusammenfassung	635

22	Next.js – Fullstack-React – Grundlagen	637
22.1	Next.js – die Hintergründe	638
22.2	Installation	638
22.2.1	Die Applikation starten	641
22.3	Die Struktur einer Next.js-Applikation	641
22.4	React Server Components	643
22.4.1	Implementierung einer Server Component	644
22.4.2	Fehlerbehandlung in Server Components	647
22.5	Statisches vs. dynamisches Rendern	648
22.5.1	Wann rendert Next.js dynamisch?	648
22.6	Statische Generierung	651
22.7	Der App Router	652
22.7.1	Layouts	652
22.7.2	Dynamische Routensegmente	653
22.7.3	Dynamische Routensegmente statisch rendern	655
22.7.4	Umleitungen	657
22.7.5	Route Groups	659
22.7.6	Parallel Routes	662
22.7.7	Intercepting Routes	663
22.7.8	Route Handlers	669
22.8	Proxy	672
22.9	Zusammenfassung	673
23	Verbesserung der User-Experience mit Next.js	675
23.1	Client Components in Next.js	675
23.1.1	Verschachtelung von Client und Server Components	680
23.2	Arbeiten mit Suchparametern	680
23.2.1	Suchparameter in Server Components	681
23.2.2	Suchparameter in Client Components	683
23.3	»loading.tsx« – Ladezustände abbilden	685
23.4	»error.tsx« – Fehler abfangen	688
23.5	»not-found.tsx« – Anzeige bei fehlenden Daten	692
23.6	Caching in einer Next.js-Applikation	694
23.6.1	»fetch«-Cache	694
23.6.2	Full Route Cache	697
23.6.3	Router Cache	698

23.7	Revalidierung und Datenaktualisierung	698
23.7.1	Zeitgesteuerte Invalidierung	699
23.7.2	Pfadbasierte Invalidierung	700
23.7.3	Tag-basierte Invalidierung	703
23.8	Streaming	704
23.9	Cache Components	707
23.9.1	Die neuen APIs der Cache Components	709
23.9.2	Statische und dynamische Inhalte in einer Cache Component	711
23.10	Zusammenfassung	714
24	Server Functions in Next.js	717
24.1	Server Functions definieren und ausführen	717
24.2	Server Functions an Client Components weitergeben	720
24.3	Server Functions in Formularen	722
24.4	Hooks für die Arbeit mit Formularen	724
24.4.1	Den Pending-Zustand mit »useFormStatus« abdecken	724
24.4.2	Formulare mit dem »useActionState« absenden	726
24.5	Das Zusammenspiel von Server Functions und React Hook Form	731
24.6	Mit Transitionen arbeiten	734
24.7	Die »use()«-Funktion in Next.js	736
24.8	Zusammenfassung	738
25	Weitere Optimierungen in Next.js	739
25.1	Umgang mit Metadaten	739
25.1.1	Statische Metadaten	740
25.1.2	Dynamische Metadaten	742
25.1.3	Favicon, Open Graph Images und weitere dateibasierte Metadaten	743
25.2	Optimierung von Schriftdateien	746
25.3	Bilder optimieren	747
25.4	Prefetching von Links	750
25.5	Zusammenfassung	752

26 Künstliche Intelligenz in React-Applikationen	753
26.1 Architektur einer KI-Applikation	754
26.2 Modell- und Backend-Setup	756
26.3 Kommunikation zwischen React und dem LLM	757
26.3.1 Kommunikation mit einem LLM	757
26.3.2 Streaming von KI-Kommunikation in React	761
26.4 Tools in React-Applikationen aufrufen	764
26.5 Personalisierung von UIs mit KI	769
26.6 Zusammenfassung	773
Index	775