

|                                              |           |
|----------------------------------------------|-----------|
| <b>Vorwort</b> .....                         | <b>15</b> |
| <b>1 Projekt-Setup</b> .....                 | <b>21</b> |
| Visual Studio Code .....                     | 21        |
| Angular CLI .....                            | 23        |
| Node.js und Angular CLI installieren .....   | 23        |
| Ein Projekt mit der CLI generieren .....     | 23        |
| Angular-Anwendung starten .....              | 24        |
| Build mit CLI .....                          | 26        |
| Projektstruktur von CLI-Projekten .....      | 27        |
| Internet Explorer 11 .....                   | 31        |
| Eine Style-Bibliothek installieren .....     | 32        |
| Alternativen zu Bootstrap .....              | 34        |
| Zusammenfassung .....                        | 35        |
| <b>2 Erste Schritte mit TypeScript</b> ..... | <b>37</b> |
| Motivation .....                             | 37        |
| Mit TypeScript starten .....                 | 38        |
| Hallo Welt! .....                            | 38        |
| Variablen deklarieren .....                  | 39        |
| Ausgewählte Datentypen in TypeScript .....   | 41        |
| Ein erstes Objekt samt Modul .....           | 44        |
| Auto-Importe mit Visual Studio Code .....    | 47        |
| Klassen .....                                | 47        |
| Funktionen und Lambda-Ausdrücke .....        | 50        |
| Interfaces und Vererbung .....               | 53        |
| Interfaces .....                             | 53        |
| Klassenvererbung .....                       | 56        |
| Type Assertion (Type Casting) .....          | 58        |

|                                                              |           |
|--------------------------------------------------------------|-----------|
| Abstrakte Klassen . . . . .                                  | 59        |
| Zugriff auf die Basisklasse . . . . .                        | 60        |
| Ausgewählte Sprachmerkmale . . . . .                         | 61        |
| Getter und Setter . . . . .                                  | 61        |
| Generics . . . . .                                           | 62        |
| Exceptions . . . . .                                         | 64        |
| Spread-Operator . . . . .                                    | 66        |
| Strikte Null-Prüfungen . . . . .                             | 66        |
| Asynchrone Operationen . . . . .                             | 68        |
| Callbacks und die Pyramide of Doom . . . . .                 | 69        |
| Promises . . . . .                                           | 70        |
| async und await . . . . .                                    | 71        |
| Bedeutung von Promises in Angular . . . . .                  | 72        |
| Zusammenfassung . . . . .                                    | 72        |
| <b>3 Eine erste Angular-Anwendung . . . . .</b>              | <b>73</b> |
| Angular-Komponente erzeugen . . . . .                        | 74        |
| Komponentenlogik . . . . .                                   | 75        |
| Auf das Backend zugreifen . . . . .                          | 77        |
| Templates und die Datenbindung . . . . .                     | 82        |
| Two-Way-Binding . . . . .                                    | 82        |
| Property-Bindings . . . . .                                  | 84        |
| Direktiven . . . . .                                         | 84        |
| Pipes . . . . .                                              | 86        |
| Event-Bindings . . . . .                                     | 86        |
| Das gesamte Template . . . . .                               | 87        |
| Komponenten einbinden . . . . .                              | 88        |
| Anwendung ausführen und debuggen . . . . .                   | 89        |
| Anwendung starten . . . . .                                  | 89        |
| Fehler in der Entwicklerkonsole entdecken . . . . .          | 90        |
| Die Anwendung im Browser debuggen . . . . .                  | 91        |
| Debuggen mit Visual Studio Code . . . . .                    | 92        |
| Zusammenfassung . . . . .                                    | 94        |
| <b>4 Komponenten und Datenbindung . . . . .</b>              | <b>95</b> |
| Datenbindung in Angular . . . . .                            | 95        |
| Rückblick auf AngularJS 1.x . . . . .                        | 95        |
| Property-Binding . . . . .                                   | 96        |
| Event-Bindings . . . . .                                     | 97        |
| Das Zusammenspiel von Property- und Event-Bindings . . . . . | 98        |
| Bindings im Template . . . . .                               | 99        |
| Two-Way-Bindings . . . . .                                   | 99        |

|                                                        |            |
|--------------------------------------------------------|------------|
| Eigene Komponenten mit Datenbindung .....              | 100        |
| Überblick .....                                        | 101        |
| Vorbereitungen .....                                   | 101        |
| Eine Komponente mit Property-Bindings .....            | 103        |
| Komponenten mit Event-Bindings .....                   | 108        |
| Komponenten mit Two-Way-Bindings .....                 | 111        |
| Life-Cycle-Hooks .....                                 | 112        |
| Ausgewählte Hooks .....                                | 112        |
| Experiment mit Life-Cycle-Hooks .....                  | 113        |
| Angular und Zyklen .....                               | 115        |
| DateControl mit Life-Cycle-Hooks .....                 | 117        |
| Zusammenfassung .....                                  | 121        |
| <b>5 Services und Dependency Injection .....</b>       | <b>123</b> |
| Ein erster Service .....                               | 123        |
| Services austauschen .....                             | 128        |
| Services mit klassischen Providern konfigurieren ..... | 133        |
| Einen Service lokal registrieren .....                 | 135        |
| Arten von Providern .....                              | 137        |
| useClass .....                                         | 138        |
| useValue .....                                         | 138        |
| useFactory .....                                       | 140        |
| useExisting .....                                      | 141        |
| multi .....                                            | 142        |
| Konstanten als Tokens .....                            | 145        |
| Zusammenfassung .....                                  | 147        |
| <b>6 Pipes .....</b>                                   | <b>149</b> |
| Überblick .....                                        | 149        |
| Built-in-Pipes .....                                   | 149        |
| Eigene Pipes .....                                     | 150        |
| Pure Pipes .....                                       | 151        |
| Implementierung einer einfachen Pipe .....             | 151        |
| Pipes registrieren und nutzen .....                    | 154        |
| Weiterführende Konstellationen .....                   | 155        |
| Pipes und Objekte .....                                | 155        |
| Pipes und Direktiven .....                             | 157        |
| Pipes und Services .....                               | 158        |
| Aufräumarbeiten mit ngOnDestroy .....                  | 159        |
| Zusammenfassung .....                                  | 160        |

|          |                                                           |            |
|----------|-----------------------------------------------------------|------------|
| <b>7</b> | <b>Module</b>                                             | <b>161</b> |
|          | Motivation                                                | 161        |
|          | Eine Angular-typische Modulstruktur                       | 162        |
|          | Shared Modules                                            | 163        |
|          | Feature-Modules                                           | 167        |
|          | Root-Modules                                              | 169        |
|          | Module reexportieren                                      | 170        |
|          | Zusammenfassung                                           | 172        |
| <b>8</b> | <b>Routing</b>                                            | <b>173</b> |
|          | Überblick                                                 | 173        |
|          | Erste Schritte mit dem Router                             | 175        |
|          | Routing-Konfiguration für das AppModule einrichten        | 176        |
|          | Routing-Konfiguration für Feature-Modules einrichten      | 178        |
|          | Platzhalter in AppComponent hinterlegen                   | 179        |
|          | Hyperlinks zum Aktivieren von Routen einrichten           | 180        |
|          | Parametrisierte Routen                                    | 182        |
|          | Arten von Routing-Parametern                              | 182        |
|          | Parameter in Komponenten auslesen                         | 182        |
|          | Parametrisierte Routen konfigurieren                      | 184        |
|          | Auf parametrisierte Routen verweisen                      | 185        |
|          | Hierarchisches Routing mit Child-Routes                   | 186        |
|          | Überblick über Child-Routes                               | 186        |
|          | Child-Komponente implementieren                           | 187        |
|          | Child-Komponente registrieren                             | 189        |
|          | Hyperlinks zum Aktivieren von Child-Routen einrichten     | 190        |
|          | Aux-Routes                                                | 191        |
|          | Platzhalter für Aux-Routes                                | 192        |
|          | Komponente für Aux-Route erzeugen                         | 193        |
|          | Konfiguration für Aux-Route                               | 194        |
|          | Verweise auf Aux-Routes einrichten                        | 195        |
|          | Mit dem Query-String und dem Hash-Fragment arbeiten       | 196        |
|          | QueryString und Hash-Fragment programmatisch beeinflussen | 197        |
|          | Query-String und Hash-Fragment deklarativ beeinflussen    | 199        |
|          | Query-String und Hash-Fragment auslesen                   | 200        |
|          | HTML5-Routing vs. Hash-Routing                            | 201        |
|          | PathLocationStrategy                                      | 201        |
|          | HashLocationStrategy                                      | 203        |
|          | Zusammenfassung                                           | 203        |

|           |                                                          |            |
|-----------|----------------------------------------------------------|------------|
| <b>9</b>  | <b>Template-getriebene Formulare und Validierung</b>     | <b>205</b> |
|           | FormsModule einbinden                                    | 206        |
|           | Eingaben validieren                                      | 206        |
|           | Zugriff auf den Zustand des Formulars                    | 207        |
|           | Bedingte Formatierung von Eingabefeldern                 | 212        |
|           | Eigene Validierungsdirektiven                            | 212        |
|           | Eine erste Validierungsdirektive erstellen               | 213        |
|           | Parametrisierbare Validierungsdirektiven                 | 218        |
|           | Multi-Field-Validatoren erstellen                        | 221        |
|           | Asynchrone Validatoren                                   | 223        |
|           | Komponente zum Präsentieren von Validierungsfehlern      | 226        |
|           | Die Standardsteuerelemente von HTML nutzen               | 228        |
|           | Checkboxes                                               | 229        |
|           | Radiobuttons                                             | 229        |
|           | Drop-down-Felder                                         | 230        |
|           | Zusammenfassung                                          | 231        |
| <b>10</b> | <b>Reaktive Formulare</b>                                | <b>233</b> |
|           | Erste Schritte mit reaktiven Formularen                  | 233        |
|           | Modul einbinden                                          | 233        |
|           | Das Formular mit einem Objektgraphen beschreiben         | 234        |
|           | Reaktive Formulare mit dem FormBuilder beschreiben       | 236        |
|           | Einen Objektgraphen an ein Formular binden               | 237        |
|           | Werte ins Formular schreiben                             | 239        |
|           | Validatoren                                              | 239        |
|           | Synchrone Validatoren                                    | 240        |
|           | Parametrisierte Validatoren                              | 242        |
|           | Asynchrone Validatoren                                   | 243        |
|           | Multi-Field-Validatoren für reaktive Formulare           | 246        |
|           | Geschachtelte Formulare                                  | 248        |
|           | Geschachtelte FormGroup's                                | 248        |
|           | Wiederholgruppen mit FormArray                           | 250        |
|           | Dynamische Formulare                                     | 253        |
|           | Zusammenfassung                                          | 254        |
| <b>11</b> | <b>Reactive Extensions Library for JavaScript (RxJS)</b> | <b>255</b> |
|           | Grundlegende Typen von RxJS                              | 255        |
|           | Observables, Observer und Operatoren                     | 255        |
|           | Observables instanziiieren                               | 258        |
|           | Subjects                                                 | 261        |

|                                          |            |
|------------------------------------------|------------|
| Observables vs. Promises .....           | 262        |
| Observables in Promises umwandeln .....  | 263        |
| Promises in Observables umwandeln .....  | 263        |
| Gruppen von Operatoren .....             | 264        |
| Creation Operators .....                 | 264        |
| Transformation Operators .....           | 265        |
| Filtering Operators .....                | 265        |
| Join Operators .....                     | 266        |
| Error Handling Operators .....           | 267        |
| Multicasting Operators .....             | 267        |
| Utility Operators .....                  | 267        |
| Reaktiver Entwurf .....                  | 267        |
| Flattening .....                         | 271        |
| Datenflüsse kombinieren .....            | 272        |
| Der Operator combineLatest .....         | 272        |
| combineLatest vs. withLatestFrom .....   | 275        |
| Der Operator merge .....                 | 277        |
| Multicasting .....                       | 279        |
| Motivation für Multicasting .....        | 279        |
| Hot vs. Cold Observables .....           | 281        |
| Fehlerbehandlung .....                   | 282        |
| Observables schließen .....              | 285        |
| Reaktive Services .....                  | 287        |
| Zusammenfassung .....                    | 291        |
| <b>12 Testautomatisierung .....</b>      | <b>293</b> |
| Jasmine und Karma .....                  | 293        |
| Aufbau eines Jasmine-Tests .....         | 293        |
| Tests mit Karma ausführen .....          | 295        |
| Karma auf dem Build-Server .....         | 296        |
| Angular und Jasmine .....                | 297        |
| Komponenten mit dem TestBed testen ..... | 297        |
| Arbeiten mit Attrappen (Mocks) .....     | 299        |
| Gray-Box-Tests mit Spys .....            | 304        |
| HTTP-Zugriffe simulieren .....           | 306        |
| Asynchrone Tests .....                   | 308        |
| Templates mit DOM-Zugriffen testen ..... | 310        |
| Direktiven testen .....                  | 312        |
| Pipes testen .....                       | 312        |
| Testabdeckung ermitteln .....            | 313        |
| Zusammenfassung .....                    | 314        |

|                                                                  |            |
|------------------------------------------------------------------|------------|
| <b>13 Performancetuning</b>                                      | <b>315</b> |
| Optimierte Datenbindung mit OnPush                               | 315        |
| Datenbindung visualisieren                                       | 316        |
| Immutables                                                       | 317        |
| Immutables und Datenbindung                                      | 319        |
| Observables und Datenbindung                                     | 319        |
| Immutables und/oder Observables                                  | 324        |
| Manuelle Änderungsverfolgung                                     | 324        |
| Lazy Loading von Routen                                          | 325        |
| Routen für das Lazy Loading einrichten                           | 325        |
| Lazy Loading im Browser nachvollziehen                           | 327        |
| Lazy Loading und Tree-Shakable Provider                          | 328        |
| Lazy Loading, klassische Provider und Shared Modules             | 330        |
| Korrekte Nutzung von SharedModules beim Einsatz von Lazy Loading | 335        |
| Preloading                                                       | 337        |
| Preloading aktivieren                                            | 337        |
| Eigene Preloading-Strategien entwickeln                          | 339        |
| Selektives Preloading mit eigener Preloading-Strategie           | 340        |
| Zusammenfassung                                                  | 341        |
| <b>14. Querschnittsfunktionen</b>                                | <b>343</b> |
| Guards                                                           | 343        |
| Das Aktivieren von Routen verhindern                             | 344        |
| Das Deaktivieren einer Komponente verhindern                     | 346        |
| Events                                                           | 349        |
| Resolver                                                         | 351        |
| Vorbereitungen                                                   | 351        |
| Resolver erzeugen und verwenden                                  | 352        |
| HttpInterceptoren                                                | 355        |
| Zusammenfassung                                                  | 358        |
| <b>15. Authentifizierung und Autorisierung</b>                   | <b>359</b> |
| Cookie-basierte Security                                         | 359        |
| Cookies und XSRF                                                 | 360        |
| Tokenbasierte Security                                           | 361        |
| OAuth 2 und OpenID Connect                                       | 361        |
| OAuth 2                                                          | 361        |
| Benutzer mit OpenID Connect authentifizieren                     | 363        |
| JSON Web Token                                                   | 364        |
| OAuth-2- und OIDC-Flows                                          | 365        |
| OAuth 2 und OIDC mit Angular nutzen                              | 366        |
| Zusammenfassung                                                  | 370        |

|                                                            |                |
|------------------------------------------------------------|----------------|
| <b>16 Internationalisierung</b>                            | <b>371</b>     |
| I18N mit dem Angular-Compiler                              | 371            |
| Überblick                                                  | 372            |
| @angular/localize installieren                             | 373            |
| Texte markieren                                            | 373            |
| Strings in der Komponentenklasse markieren                 | 374            |
| Texte extrahieren                                          | 375            |
| Übersetzte Texte in Builds integrieren                     | 377            |
| Sprache beim Einsatz von ng serve festlegen                | 379            |
| Übersetzungstexte zur Laufzeit angeben                     | 380            |
| Grammatikalische Formen berücksichtigen                    | 382            |
| Unterschiedliche Formate unterstützen                      | 383            |
| Manuell weitere Formate laden                              | 384            |
| ngx-translate                                              | 385            |
| Überblick                                                  | 385            |
| Bibliothek installieren und konfigurieren                  | 386            |
| Sprachdateien bereitstellen                                | 388            |
| Texte einbinden                                            | 388            |
| Texte zur Laufzeit laden                                   | 389            |
| Sprachwechsel                                              | 390            |
| Grammatikalische Formen berücksichtigen                    | 391            |
| Unterschiedliche Formate nutzen                            | 393            |
| Lazy Loading                                               | 393            |
| Zusammenfassung                                            | 394            |
| <br><b>17 Reaktive Zustandsverwaltung mit NGRX (Redux)</b> | <br><b>397</b> |
| Zustandsverwaltung mit Services                            | 397            |
| Das Redux-Muster                                           | 398            |
| NGRX einrichten                                            | 400            |
| Building-Blocks implementieren                             | 402            |
| State modellieren                                          | 402            |
| Actions festlegen                                          | 405            |
| Reducer definieren                                         | 405            |
| Effect einrichten                                          | 406            |
| Auf den Store zugreifen                                    | 407            |
| Debuggen mit dem Store                                     | 408            |
| Selektoren                                                 | 410            |
| Ein erster Selektor                                        | 410            |
| Selektoren verschachteln                                   | 411            |
| Meta-Reducer                                               | 412            |
| Zusammenfassung                                            | 413            |



**18 Details zu Komponenten und Direktiven . . . . . 415**

Vorbereitungen . . . . . 415

Weiterführende Aspekte von Komponenten. . . . . 418

    Content Projection . . . . . 418

    Parent-Komponenten referenzieren . . . . . 420

    View und Content. . . . . 424

    Kommunikation über Template-Variablen. . . . . 434

    Kommunikation über Services . . . . . 434

Attributdirektiven . . . . . 438

    Direktiven definieren . . . . . 438

    Mit der Umwelt kommunizieren . . . . . 440

    Direktiven und Template-Variablen. . . . . 442

Strukturelle Direktiven . . . . . 443

    Templates und Container. . . . . 443

    Microsyntax . . . . . 445

    Eine einfache DataTable umsetzen. . . . . 447

    ViewContainerRef direkt zum Einblenden von Templates  
    verwenden. . . . . 452

    Bestehende ViewContainer ergänzen . . . . . 453

    Dialoge dynamisch einblenden. . . . . 454

    ViewContainerRef direkt zum dynamischen Erzeugen von  
    Komponenten verwenden. . . . . 462

Komponenten für Formularfelder . . . . . 463

Ausgaben formatieren und Eingaben parsen. . . . . 464

Eigene Formularsteuerelemente . . . . . 467

Zusammenfassung. . . . . 470

**19 Wiederverwendbare Bibliotheken und Monorepos . . . . . 471**

Monorepo erstellen . . . . . 471

Aufbau von Bibliotheken. . . . . 473

Bibliothek in Monorepo ausprobieren. . . . . 475

npm-Paket bauen und bereitstellen. . . . . 478

npm-Paket konsumieren . . . . . 480

Zusammenfassung. . . . . 480

**Index . . . . . 481**