

<b>Vorwort zur 3. Auflage</b> . . . . .	<b>XXIII</b>
Warum dieses Buch? . . . . .	XXIII
Was ist der Inhalt dieses Buchs? . . . . .	XXIV
Welche Voraussetzungen sollten Sie für dieses Buch mitbringen? . . . . .	XXV
Was ist das Ziel des Buchs? . . . . .	XXV
Welche Versionen werden in diesem Buch behandelt? . . . . .	XXVI
Welche Programmiersprache wird in diesem Buch verwendet? . . . . .	XXVII
Welche Werkzeuge werden in diesem Buch verwendet? . . . . .	XXVII
Welche Sprachversion wird in diesem Buch verwendet? . . . . .	XXVII
Woher bekommen Sie die Code-Beispiele und die drei Bonuskapitel zu diesem Buch? . . . . .	XXVIII
Wem ist zu danken? . . . . .	XXX
Die Autoren . . . . .	XXX
<b>Teil A Web-Basiswissen</b> . . . . .	<b>1</b>
<b>1 Protokolle, Standards und Konzepte</b> . . . . .	<b>3</b>
Standardisierung . . . . .	3
RFCs . . . . .	3
World Wide Web Consortium (W3C) . . . . .	4
European Computer Manufacturers Association (ECMA) . . . . .	4
Hypertext Transfer Protocol (HTTP) . . . . .	4
Protokollaufbau, Header, Body . . . . .	5
Kommandoaufbau . . . . .	6
Die HTTP-Statuscodes . . . . .	7
Ablauf einer HTTP-Kommunikation . . . . .	7
Kopffelder . . . . .	8
HTTP 2.0 . . . . .	9
Ergänzende Standards zu HTTP . . . . .	10
Web-Sockets . . . . .	10
WebDAV . . . . .	11

Simple Mail Transfer Protocol (SMTP) oder Extended SMTP (ESMTP) . . . . .	11
File Transfer Protocol (FTP) . . . . .	11
REpresentational State Transfer (REST) . . . . .	12
Webdienste . . . . .	12
Merkmale. . . . .	12
REST-Beispiel . . . . .	13
URI . . . . .	14
HTTP-Methoden für REST . . . . .	15
Multipurpose Internet Mail Extensions (MIME). . . . .	16
JavaScript Object Notation (JSON). . . . .	17
Atom Syndication Format (ATOM) . . . . .	18
Grenzen von REST: GraphQL und OData . . . . .	19
Single Page Application (SPA). . . . .	19
Asynchronous JavaScript and XML (AJAX) . . . . .	20
Responsive Web Design (RWD). . . . .	21
<b>2 Hypertext Markup Language (HTML) . . . . .</b>	<b>23</b>
Grundlagen HTML . . . . .	23
Geschichte . . . . .	23
XML – Grundlage für HTML . . . . .	24
Markup . . . . .	25
XML-Dokumente. . . . .	25
Verarbeitung . . . . .	26
Der Begriff »Markup« . . . . .	26
Seitenaufbau . . . . .	29
Document Object Model (DOM) . . . . .	30
Der Dokumenttyp (Doctype). . . . .	30
Codierungen . . . . .	30
HTML5-Seitenstruktur . . . . .	31
Elemente der Seite. . . . .	34
Textelemente . . . . .	34
Fließtextauszeichnung . . . . .	36
Verweise. . . . .	37
Tabellen . . . . .	37
Multimedia und Grafiken . . . . .	38
Formulare . . . . .	38
Skripte . . . . .	39
Interaktive Elemente . . . . .	39
Allgemeine und universelle Attribute . . . . .	39
id . . . . .	40
class . . . . .	40
accesskey . . . . .	41
contenteditable . . . . .	41
contextmenu . . . . .	42

dir . . . . .	42
draggable . . . . .	42
dropzone. . . . .	43
hidden. . . . .	43
lang . . . . .	43
spellcheck . . . . .	43
style. . . . .	44
tabindex . . . . .	44
title . . . . .	44
<b>3 Cascading Style Sheets (CSS) . . . . .</b>	<b>45</b>
Grundlagen . . . . .	45
Syntax . . . . .	46
Selektor . . . . .	47
Elemente (Tags) . . . . .	47
ID . . . . .	47
Klassen . . . . .	47
Attribute . . . . .	48
Logische Auswahl. . . . .	48
Weitere Selektoren . . . . .	48
Das Box-Modell. . . . .	51
Bausteine der Box. . . . .	52
Zählweise . . . . .	53
Ausnahmen. . . . .	53
Das Box-Modell in CSS3 . . . . .	53
Das Flexbox-Modell . . . . .	53
Prinzipien . . . . .	53
Eigenschaft der Container . . . . .	55
Beachtung von Medien . . . . .	56
Syntax . . . . .	56
Parameter . . . . .	59
Der Viewport . . . . .	63
Viewport einstellen. . . . .	63
Parameter für den Viewport . . . . .	65
Einheiten . . . . .	65
Absolute Einheiten . . . . .	66
Relative Einheiten. . . . .	67
<b>4 Bootstrap . . . . .</b>	<b>69</b>
Einführung in Bootstrap . . . . .	69
Neuerungen in Bootstrap 4 . . . . .	69
Installation . . . . .	70
Struktur der CSS-Dateien . . . . .	72
Seitenaufbau . . . . .	73
Browserunterstützung . . . . .	74

ARIA . . . . .	75
Optimierung . . . . .	76
Hilfsklassen . . . . .	78
Reboot . . . . .	78
Struktur der Seite . . . . .	79
Einführung. . . . .	79
Das Rastersystem. . . . .	80
Das Flex-Raster . . . . .	89
Typografie. . . . .	94
Überschriften . . . . .	95
Text und Textelemente . . . . .	96
Listen . . . . .	101
Tabellen . . . . .	103
Hilfsklassen . . . . .	112
Formulare . . . . .	114
Struktur eines Formulars . . . . .	114
Eingabeelemente . . . . .	124
Schaltflächen . . . . .	139
Weitere Bausteine . . . . .	144
Symbole . . . . .	144
Responsive Bilder. . . . .	148
Eingebettete Quellen . . . . .	149
Farben und Hintergründe . . . . .	149
Ausrichtung von Elementen im Fluss . . . . .	151
Inhalte anzeigen und verstecken . . . . .	152
Komponenten . . . . .	152
Klappmenü (dropdown). . . . .	153
Werkzeuggestreife (toolbar) . . . . .	156
Schaltfläche mit Menü (button group) . . . . .	159
Navigation (nav, navbar) . . . . .	162
Pfadnavigation (breadcrumb) . . . . .	170
Seitenweises Blättern (pagination) . . . . .	171
Kennzeichnungen (tag) . . . . .	173
Großbildleinwand (jumbotron) . . . . .	175
Seitenüberschriften (page header) . . . . .	176
Meldungen (alert) . . . . .	177
Fortschrittsbalken (progress) . . . . .	178
Medien (media) . . . . .	182
Allgemeine Listen (list group) . . . . .	183
Karten (cards) . . . . .	187
Aktive Komponenten . . . . .	197
Einrichtung und Aktivierung. . . . .	197
Die Programmierschnittstelle . . . . .	198
Übergänge (transition) . . . . .	199
Modale Dialoge (modals) . . . . .	200

Klappmenü (dropdown) . . . . .	208
Scrollbar-Überwachung (scrollspy) . . . . .	209
Angeheftete Navigation (affix) . . . . .	211
Umschaltbare Tabulatoren (tab) . . . . .	211
Tooltips (tooltip) . . . . .	213
Inhaltsüberlagerung (popover) . . . . .	216
Meldungen (alert) . . . . .	219
Interaktive Schaltflächen (button) . . . . .	220
Inhaltseinblendung (collapse) . . . . .	221
Bilderkreis (carousel) . . . . .	225

## **Teil B Webserverprogrammierung mit ASP.NET . . . . . 229**

<b>5 ASP.NET MVC . . . . .</b>	<b>231</b>
Geschichte und Verbreitung von ASP.NET . . . . .	231
Architektur von ASP.NET MVC . . . . .	232
Erste Schritte mit ASP.NET MVC . . . . .	233
ASP.NET-MVC-Projekt anlegen . . . . .	233
Projektaufbau . . . . .	235
Nugget-Pakete . . . . .	236
Controller anlegen . . . . .	238
View anlegen . . . . .	240
Webanwendung testen . . . . .	245
Einträge editieren . . . . .	246
Vorschlagswerte über Drop-down-Listenfelder anbieten . . . . .	251
Controller . . . . .	254
Models entgegennehmen . . . . .	254
View auswählen . . . . .	256
Auf Ausnahmen reagieren . . . . .	257
URL-Mapping beeinflussen (Routing) . . . . .	257
Asynchrone Controller . . . . .	260
Vereinfachte Implementierung asynchroner Controller seit .NET 4.5 . . . . .	261
Views . . . . .	262
Razor . . . . .	262
Razor-Helper . . . . .	265
Layoutseiten . . . . .	265
Partielle Views . . . . .	267
Vorlagen für Felder und Models . . . . .	269
Views für mobile Anwendungen . . . . .	272
Zwischen unterschiedlichen Ansichten wechseln . . . . .	273
Minification und Bundling . . . . .	274

Models	275
Metadaten im Model festlegen	275
HTML-Text übermitteln	276
Validieren von Benutzereingaben	276
Globalisierung	282
Sprach- und Ländereinstellungen festlegen	282
Über Ressourcdateien mehrsprachige Inhalte bereitstellen	284
Codierung festlegen	285
Areas	286
Filter	288
Überblick	289
Umsetzung	289
Filter auf Controller und Action-Methoden anwenden	291
Globale Filter	292
Authentifizierungsfilter	292
<b>6 ASP.NET Web API</b>	<b>295</b>
Einen einfachen HTTP-Service erstellen	295
Parameter und Rückgabewerte	296
HTTP-Services konfigurieren	298
HTTP-Services mit Fiddler testen	300
Mehr Kontrolle über HTTP-Nachrichten	301
Antworten mit HttpResponseMessage beeinflussen	301
Anfragen als HttpRequestMessage darstellen	303
HttpRequestMessage und HttpResponseMessage am Beispiel	
Conditional Get	304
Antworten über IHttpActionResult zurückgeben	305
HTTP-Services über HttpClient konsumieren	306
Routing	309
Benutzerdefinierte Routen über die Konfiguration festlegen	309
Attributbasiertes Routing	310
Weiterführende Schritte mit der Web API	313
Dynamische Parameter	314
Tracing	315
Request Batching	316
Cross Origin Resource Sharing (CORS)	318
Validieren	319
Querschnittsfunktionen implementieren	320
Message-Handler	320
Filter	323
Filterüberschreibungen	330
Benutzerdefinierte Formate unterstützen	330
Formatter implementieren	331
Formatter mit HttpClient verwenden	333

Serialisierung beeinflussen . . . . .	334
JSON-Serializer konfigurieren . . . . .	334
XML-Serializer konfigurieren . . . . .	335
Eigenschaften von der Serialisierung ausschließen. . . . .	335
Zirkuläre Referenzen serialisieren. . . . .	336
Binary JSON (BSON) . . . . .	338
Web API und HTML-Formulare . . . . .	338
Einfache Formularfelder übermitteln . . . . .	338
Datei-Upload via HTML-Formular . . . . .	339
Fortschritt ermitteln. . . . .	341
Feingranulare Konfiguration . . . . .	343
Controllerbasierte Konfiguration . . . . .	343
Routenbasierte Konfiguration . . . . .	344
<b>7 ASP.NET SignalR . . . . .</b>	<b>345</b>
Long-Polling . . . . .	345
Web-Sockets . . . . .	346
Überblick über ASP.NET SignalR . . . . .	347
PersistentConnection. . . . .	348
Erste Schritte mit SignalR und PersistentConnection . . . . .	348
Lifecycle-Methoden . . . . .	349
URL-Mapping für persistente Verbindungen. . . . .	349
Einfacher Client für eine persistente Verbindung. . . . .	350
Einfacher JavaScript-Client für eine persistente Verbindung. . . . .	351
Hubs . . . . .	353
Methoden und Callbacks mit SignalR und Hubs . . . . .	353
URL-Mapping für Hubs . . . . .	355
Lifecycle-Methoden . . . . .	355
Hubs konsumieren . . . . .	355
Hubs über JavaScript konsumieren . . . . .	357
Gruppen . . . . .	361
Pipeline-Module für Querschnittsfunktionen . . . . .	362
SignalR konfigurieren . . . . .	364
Cross Origin Resource Sharing (CORS) . . . . .	364
SignalR skalieren . . . . .	364
Überlegungen zum Skalieren von SignalR . . . . .	365
SignalR mit SQL Server skalieren . . . . .	366
Implementierung eines SignalR-Clients . . . . .	368
Das Skalierungsszenario testen. . . . .	369
Azure Service Bus und Redis als Alternative zu SQL Server. . . . .	370

<b>8</b>	<b>ASP.NET-Programmierschnittstellen</b> .....	<b>371</b>
	Direkt mit HTTP interagieren.....	371
	HttpContext .....	371
	Server (HttpServerUtility) .....	373
	Request (HttpRequest) .....	374
	Response (HttpResponse) .....	376
	Zustandsverwaltung auf Sitzungsebene .....	377
	Überblick.....	378
	Weitere Optionen .....	379
	Programmieren mit dem Sitzungszustand .....	380
	URL-basierte Sitzungsverwaltung ohne Cookies.....	380
	Konfiguration des Sitzungszustands .....	381
	Speicherort der Sitzungstabelle wählen.....	382
	Komprimierung des Sitzungszustands.....	383
	Deaktivieren des Sitzungszustands .....	383
	Caching.....	383
	Überblick.....	384
	Pro und Contra der Zwischenspeicherung .....	384
	Zwischenspeicherung ganzer Seiten (Output-Caching) .....	385
	Caching von Seitenteilen (Fragmentzwischenspeicherung).....	387
	Programmatisches Caching .....	388
	Cacheinvalidierung .....	389

## **Teil C Webserverprogrammierung mit ASP.NET Core ..... 393**

<b>9</b>	<b>Einführung in ASP.NET Core</b> .....	<b>395</b>
	Klassisches ASP.NET oder ASP.NET Core? .....	395
	Einführung in die Core-Welt .....	396
	.NET Standard .....	397
	Windows Compatibility Pack für .NET Core.....	402
	Open Source .....	405
	Dokumentation.....	405
	Werkzeuge .....	408
	Kommandozeilenwerkzeug dotnet .....	408
	Editoren .....	410
	Erste Schritte mit ASP.NET Core (auf .NET Core) .....	410
	Installation.....	410
	Projekt anlegen .....	411
	Projektaufbau.....	414
	Die Klasse Program .....	415
	Klasse Startup und Middleware.....	418



Referenzen/Pakete . . . . .	420
Übersetzen und Debugging . . . . .	423
Deployment . . . . .	426
ASP.NET Core auf dem klassischen .NET Framework . . . . .	429
Einsatzszenarien . . . . .	429
Anlegen von ASP.NET-Core-Projekten mit dem klassischen .NET Framework . . . . .	430
Projektaufbau . . . . .	430
Referenzen/Pakete . . . . .	432
Deployment . . . . .	433
In ASP.NET Core integrierte Webserver: Kestrel versus HTTP.sys (WebListener) . . . . .	435
Kestrel . . . . .	435
HTTP.sys . . . . .	436
<b>10 ASP.NET Core MVC und Razor Pages . . . . .</b>	<b>439</b>
POCO-Controller . . . . .	440
Controller und View . . . . .	441
Tag Helper . . . . .	442
Wie funktionieren Tag Helper? . . . . .	442
Eingebaute Tag Helper . . . . .	443
Eigene Tag Helper . . . . .	444
Beispiel: Tag <autor> . . . . .	445
Beispiel: Tag <row> . . . . .	446
Beispiel: Tag <condition> . . . . .	448
Beispiel: Tag <repeater> . . . . .	448
View Components . . . . .	450
Razor Pages . . . . .	452
Von Webforms über MVC zu Razor Pages . . . . .	453
Razor Pages versus MVC . . . . .	454
Page Model als Code-Behind . . . . .	455
URL-Parameter . . . . .	456
Eingebaute Objekte . . . . .	458
Datenbindung . . . . .	459
Praxisbeispiel zu Razor Pages . . . . .	460
Drittanbieterkomponenten für ASP.NET Core . . . . .	468
<b>11 ASP.NET-Core-Klassenbibliotheken . . . . .</b>	<b>469</b>
Dependency Injection in ASP.NET Core . . . . .	470
Implementierung eines Diensts . . . . .	471
Registrierung von Diensten . . . . .	472
Injektion in einen Konstruktor . . . . .	473
Manuelle Beschaffung . . . . .	473
Injektion in eine View . . . . .	474
Vordefinierte Dienste der ASP.NET-Core-Infrastruktur . . . . .	474

Konfiguration in ASP.NET Core . . . . .	475
Sitzungen in ASP.NET Core . . . . .	477
Cookies . . . . .	480
URL-Rewriting in ASP.NET Core. . . . .	480
Benutzerverwaltung und Authentifizierung . . . . .	483
<b>12 ASP.NET Core Web APIs . . . . .</b>	<b>487</b>
Die Grundlagen von ASP.NET Core Web API . . . . .	487
Abfragen von Daten. . . . .	487
Arbeiten mit Statuscodes . . . . .	490
Anlegen, Aktualisieren und Löschen von Daten . . . . .	491
Nutzung von Sub-Routen für Teil-Ressourcen . . . . .	494
Nutzung weiterer Model-Binding-Funktionen . . . . .	495
Anbieten unterschiedlicher Repräsentationen von Ressourcen. . . . .	497
Nutzung des PATCH-Verbs mit JSON-Patch . . . . .	497
Cross Origin Resource Sharing . . . . .	499
Bereitstellung von OpenAPI-Beschreibungen über Swagger . . . . .	501
Starten von Hintergrundprozessen über IHostedService. . . . .	504
Integrationstests mithilfe des TestHosts . . . . .	505
Service-Kommunikation mit HttpClient. . . . .	506
Abfragen von Daten. . . . .	506
Versenden von Daten und Reaktion auf Statuscodes . . . . .	508
Arbeiten mit Headern und HttpRequestMessage . . . . .	509
Generieren von Clients für Swagger-Spezifikationen. . . . .	509
Ausblick auf Neuerungen durch ASP.NET Core 2.1 . . . . .	510
Annotation [ApiController]. . . . .	510
ActionResult<T>-Rückgabewerte . . . . .	511
Optimierte Input-Verarbeitung . . . . .	511
<b>13 Microservices mit ASP.NET Core Web API und RabbitMQ . . . . .</b>	<b>515</b>
Grundlagen von Microservices . . . . .	515
Mögliche Vorteile von Microservices. . . . .	517
Mögliche Nachteile von Microservices . . . . .	518
Modellierung von Microservices . . . . .	519
Integration und Kommunikation von Microservices . . . . .	520
Synchrone und asynchrone Kommunikation . . . . .	520
Messaging-basierte versus Request-Response-Kommunikation . . . . .	521
Gesteuerte (orchestriert) versus Event-getriebene Kommunikation (choreografiert). . . . .	522
.NET-Core-Technologien zur Umsetzung von Service-Kommunikation . . . . .	523
Asynchrone Service-Kommunikation über RabbitMQ . . . . .	523
Microservices-Fallstudie . . . . .	530
Umgebung und fachlicher Kontext . . . . .	531
Der HumanResourcesService. . . . .	532

Der ProjectsService . . . . .	534
Der ManagementDashboardService . . . . .	543
Der InvoicingService . . . . .	545
Fazit . . . . .	550
Nützliche Patterns und Best-Practices im Bereich Microservices . . .	550
Resilienz-Patterns . . . . .	551
Serviceübergreifender Code und serviceübergreifende Funktionen	558
<b>14 ASP.NET Core SignalR . . . . .</b>	<b>561</b>
Hub-Klassen . . . . .	561
Hub-Client mit .NET Core 2.1 . . . . .	563
Hub-Client mit Angular . . . . .	564
Serverseitig initiierte Benachrichtigungen und Gruppen . . . . .	569
 <b>Teil D Web-Client-Programmierung . . . . .</b>	 <b>575</b>
<b>15 JavaScript-Grundlagen . . . . .</b>	<b>577</b>
Grundlagen der Sprache . . . . .	577
Sprachmerkmale und Entwurfsmuster . . . . .	578
Vergleich mit Programmiersprachen . . . . .	578
JavaScript-Syntax . . . . .	578
Typen . . . . .	582
Objekte . . . . .	593
Symbole . . . . .	594
Arrays . . . . .	594
Operatoren . . . . .	596
Anweisungen – Statements . . . . .	598
Fehlerbehandlung . . . . .	604
Variablen und Scope . . . . .	606
Objektorientierung . . . . .	607
Erstellen von Objekten . . . . .	607
Klassen . . . . .	609
Statische Mitglieder . . . . .	615
Vererbung . . . . .	615
Exkurs Objekthierarchie . . . . .	619
Ableiten von internen Typen . . . . .	625
Tipps zum Umgang mit objektorientierten Techniken . . . . .	625
Globale Standardfunktionen . . . . .	626
Timer-Funktionen . . . . .	626
Prüffunktionen . . . . .	627
Konvertierungsfunktionen . . . . .	627
Module . . . . .	628
Entwurfsmuster . . . . .	628
Module . . . . .	635

Funktionen	635
Pfeilfunktionen (Lambdas)	635
Erweiterte Objekliterale	638
Destrukturierung	638
Umgang mit Argumenten	639
Generatoren und Iteratoren	640
Asynchrone Programmierung	643
Klassische asynchrone Programmierung	643
Promise	650
Set und Map	654
Set	654
Map	656
Schlüsselvergleiche	656
Iteratoren	657
WeakMap und WeakSet	658
Reguläre Ausdrücke	658
Einführung	658
Kopieren oder Konstruieren?	659
Und wie funktioniert das?	659
Gruppierungen	668
Vorwärtsreferenzen	669
Die JavaScript-Funktionen	670
Zusammenfassung	674
Reflektions-API	676
Einfache Methoden	676
Reflect	676
Erzeugerfunktionen	677
Operatoren für Schlüsselwörter	677
Dynamische Argumentlisten	678
Funktionsaufrufe	678
Proxy-Fallen	678
Zusammenfassung der Reflect-Methoden	679
Stellvertreter: Proxies	680
Einführung	680
Proxy-Fallen anwenden	681
Schemaprüfung mit Proxies	683
Entfernbare Proxies	684
Übersicht	685
<b>16 TypeScript</b>	<b>687</b>
Geschichte von TypeScript	687
Open-Source-Projekt	688
TypeScript-Compiler	688
Übersetzung von TypeScript in JavaScript	689
TypeScript Playground	690

TypeScript in Visual Studio . . . . .	691
TypeScript-Kommandozeilencompiler tsc. . . . .	694
TypeScript in Visual Studio Code . . . . .	695
TypeScript in Gulp . . . . .	696
Datentypen . . . . .	698
Arrays und Tupel. . . . .	700
Klassen. . . . .	701
Generische Klassen . . . . .	703
Strukturelle Typäquivalenz (Duck Typing) . . . . .	704
Funktionen und Lambda-Ausdrücke. . . . .	704
Dekoratoren. . . . .	706
Module und Verweise . . . . .	707
TypeScript ohne externes Modulsystem (interne Module) . . . . .	707
Externe Module . . . . .	708
Re-Export . . . . .	710
Deklarationsdateien. . . . .	710
Einbindung von Deklarationsdateien (altes Verfahren) . . . . .	711
Import von Deklarationsdateien. . . . .	711
Deklarationsdateien und tsconfig.js. . . . .	711
Mixins . . . . .	712
Mixin-Konstruktortypen . . . . .	714
Deklaratives Mischen . . . . .	715
Module mischen. . . . .	716
Nicht mögliche Vermischungen . . . . .	719
Reflektion: Metadaten per Programmcode . . . . .	719
Die Metadaten-API für Reflektion . . . . .	720
Ermitteln von Typinformationen . . . . .	720
Ermitteln von Parameterinformationen . . . . .	721
Ermitteln von Rückgabeinformationen . . . . .	721
Asynchrone Programmierung . . . . .	721
Generatoren und Iteratoren. . . . .	722
Iteratoren . . . . .	722
Generatoren . . . . .	723
Asynchrone Iteration . . . . .	723
Asynchrone Generatoren . . . . .	724
<b>17 Angular . . . . .</b>	<b>725</b>
Ziele und Architektur von Angular . . . . .	725
Browserunterstützung . . . . .	726
Veröffentlichungszyklus von Angular . . . . .	727
Dokumentation . . . . .	728
Beispielsammlung . . . . .	729
Komponenten . . . . .	731
Datenbindung und Pipes . . . . .	734
Syntaxübersichten . . . . .	736

Strukturelle Direktiven .....	736
Datenbindungssyntax .....	737
Angular-Pipes .....	737
Module .....	738
Formulare .....	739
Routing .....	742
Dienste und Dependency Injection .....	744
Animationen .....	745
Werkzeuge .....	745
JiT vs. AOT .....	746
Angular Universal .....	748
<b>18 ASP.NET Blazor .....</b>	<b>749</b>
Silverlight wurde eingestellt .....	749
Web Assembly .....	749
Architektur von Blazor .....	750
Erste Schritte mit ASP.NET Blazor .....	751
Beispielprojekt .....	752
Weitere Möglichkeiten .....	755
Ausblick .....	755
<b>Teil E Hosting von ASP.NET und ASP.NET Core .....</b>	<b>757</b>
<b>19 Internet Information Services (IIS) .....</b>	<b>759</b>
Versionsgeschichte .....	759
Kernfunktionen des IIS .....	760
Installation des IIS .....	761
Modularisierung .....	763
Skriptbasierte Installation .....	764
Integration zwischen ASP.NET und IIS .....	765
Test der Installation .....	765
IIS Express .....	766
IIS-Administration .....	767
IIS-Manager .....	768
Administration per Kommandozeile und Skript .....	769
IIS-Websites (virtuelle Webserver) .....	769
Websites erstellen .....	770
Websites erstellen per Skript .....	771
Wichtige Einstellungen für Websites .....	773
Beschränken der möglichen Clients .....	773
Authentifizierung .....	774
Transport Layer Security (TLS)/Secure Socket Layer (SSL) .....	775

Server für Nicht-HTTP-Protokolle . . . . .	777
Virtuelle Verzeichnisse im IIS . . . . .	777
IIS-Anwendungen . . . . .	779
Rahmenbedingungen einer IIS-Anwendung. . . . .	779
Anlegen einer IIS-Anwendung . . . . .	780
IIS-Anwendungspools . . . . .	781
Eigenschaften eines Anwendungspools . . . . .	781
Liste der Anwendungspools . . . . .	782
Zuordnung von Websites und IIS-Anwendungen zu Anwendungspools . . . . .	782
ASP.NET-Version. . . . .	783
Erweiterte Einstellungen für Anwendungspools . . . . .	784
Anwendungspoolidentität . . . . .	785
Wiederverwendung (Recycling) . . . . .	785
Leistungseinstellungen . . . . .	786
Zustandsüberwachung . . . . .	787
Besonderheiten für ASP.NET-Anwendungen . . . . .	787
IIS-Autostart . . . . .	787
IIS-Verarbeitungspipeline. . . . .	790
<b>20 Microsoft Azure . . . . .</b>	<b>793</b>
Azure-Konzepte . . . . .	793
Anlegen einer Subscription . . . . .	794
Anlegen einer Ressource Group. . . . .	794
Anlegen eines App Service Plans . . . . .	795
Anlegen eines Azure App Service. . . . .	796
Anlegen eines Azure App Service mit der PowerShell . . . . .	799
<b>21 Verteilen von Webanwendungen aus Visual Studio heraus . . . . .</b>	<b>801</b>
Web Deploy-Werkzeug . . . . .	801
Web Deploy in einen IIS-Webserver . . . . .	802
Web Deploy nach Azure. . . . .	804
Konfigurationstransformationen . . . . .	805
Erstellen der Transformationsdateien . . . . .	805
Syntax der Transformationsdateien . . . . .	805
Ergebnis der Transformation . . . . .	807
Continuous Integration und Continuous Delivery . . . . .	807
<b>22 Webanwendungen in Docker . . . . .</b>	<b>809</b>
Docker auf Windows. . . . .	809
Installation der Docker-Unterstützung von Microsoft . . . . .	811
Installation von Docker for Windows . . . . .	813
Ein Image beschaffen. . . . .	814
Einen Container starten. . . . .	815
Ein Visual-Studio-Container-Projekt erstellen. . . . .	816

Debugging eines Containers . . . . .	817
Verwendung des Containers . . . . .	817
Images aus Containern erstellen . . . . .	821
.NET Core-Container . . . . .	821
Container in Windows Server 2016 hosten . . . . .	824
Images verbreiten . . . . .	825
<b>Teil F Fallbeispiel: MiracleList . . . . .</b>	<b>829</b>
<b>23 Das Fallbeispiel »MiracleList« . . . . .</b>	<b>831</b>
Das Szenario . . . . .	831
Links . . . . .	834
<b>24 Das MiracleList-Backend . . . . .</b>	<b>835</b>
Architektur . . . . .	837
Entitätsklassen . . . . .	840
Entity Framework Core-Kontextklasse . . . . .	842
Lebensdauer der Kontextklasse in ASP.NET Core-Anwendungen . . . . .	843
Geschäftslogik . . . . .	844
Web API . . . . .	853
<b>25 MiracleList-Web-Client . . . . .</b>	<b>865</b>
Technikeinsatz im Web-Client . . . . .	865
Angular-CLI . . . . .	866
Webserver starten . . . . .	868
Zusatzkomponenten . . . . .	869
Proxy für REST-Dienste . . . . .	870
Daten darstellen . . . . .	871
Navigation zu den Aufgaben . . . . .	873
Datumsanzeigen . . . . .	876
Zeilenumbrüche . . . . .	878
Aufgabenstatus ändern . . . . .	878
Aufgaben anlegen . . . . .	881
Suchfunktion . . . . .	882
Komponentenbildung . . . . .	883
Schritt 1: Routing-Modul . . . . .	885
Schritt 2: Kommunikationsdienst . . . . .	885
Schritt 3: Teilaufgabenliste in SubTaskList-Komponente . . . . .	886
Schritt 4: Auslagerung in TaskView . . . . .	888
Schritt 5: Aufgaben bearbeiten . . . . .	890
Schritt 6: Integration in AppComponent . . . . .	891



Datepicker . . . . .	892
Kontextmenü verwenden . . . . .	893
Nachfragen beim Löschen . . . . .	896
Animationen . . . . .	897
Sortieren der Aufgabenliste . . . . .	898
Benutzeranmeldung . . . . .	899
Hauptmenü . . . . .	901
Testen . . . . .	902
Auslieferung . . . . .	905
<b>26 MiracleList-Electron-Client . . . . .</b>	<b>907</b>
Hybride Apps . . . . .	908
Architektur von Electron . . . . .	908
Electron-Projekt anlegen . . . . .	909
Kommunikation zwischen Main und Renderer . . . . .	918
Erweiterungen in der Webanwendung . . . . .	918
Start der Electron-Anwendung . . . . .	921
Debugging . . . . .	922
Deployment . . . . .	923
<b>27 MiracleList-Cordova-Client . . . . .</b>	<b>925</b>
Architektur von Cordova . . . . .	926
Cordova-Projekt anlegen . . . . .	927
Cordova-Anwendung starten . . . . .	929
Erweiterungen in der Webanwendung . . . . .	930
Cordova-APIs verwenden . . . . .	931
Plug-ins verwenden . . . . .	932
Responsive Web Design mit Bootstrap . . . . .	935
<b>Index . . . . .</b>	<b>937</b>