

Vorwort	XXI
1 Warum Laravel?	1
Warum ein Framework verwenden?	1
»Ich baue es einfach selbst«	2
Konsistenz und Flexibilität	2
Eine kurze Geschichte der Web- und PHP-Frameworks	2
Ruby on Rails	3
Eine Welle von PHP-Frameworks	3
Das Gute und das Schlechte an CodeIgniter	3
Laravel 1, 2 und 3	4
Laravel 4	4
Laravel 5	4
Laravel 6	5
Was ist so besonders an Laravel?	5
Die Philosophie von Laravel	5
Wie sich Laravel um die Zufriedenheit des Entwicklers verdient macht	6
Die Laravel-Community	8
Wie es funktioniert	8
Warum also Laravel?	10
2 Eine Laravel-Entwicklungsumgebung einrichten	11
Systemanforderungen	11
Composer	12
Lokale Entwicklungsumgebungen	12
Laravel Valet	12
Laravel Homestead	13

Ein neues Laravel-Projekt erstellen	14
Installation von Laravel mit dem Laravel-Installationsprogramm	14
Installation von Laravel mit dem create-project-Feature von Composer	14
Das Helfer-Paket installieren	15
Lambo: »laravel new« mit Düsenantrieb	15
Die Verzeichnisstruktur von Laravel.	15
Die Ordner	16
Die Dateien	17
Konfiguration	18
Die .env-Datei	19
Achtung, fertig, los!	21
Testen	22
TL;DR.	22
3 Routing und Controller	23
Eine kurze Einführung in MVC, HTTP-Verben und REST	23
Was ist MVC?	23
Die HTTP-Verben	24
Was ist REST?	25
Routendefinitionen	26
Routing-Verben	28
Routen-Handling	28
Routenparameter	29
Benannte Routen	30
Routen gruppieren	33
Middleware	33
Pfad-Präfixe	36
Fallback-Routen	36
Subdomain-Routing.	36
Namensraum-Präfixe	37
Namenspräfixe	37
Signierte Routen	38
Eine Route signieren	38
Signierte Links zulassen	39
Views.	40
Einfache Routen direkt mit Route::view() zurückgeben	41
Verwendung von View Comosern, um Variablen für alle Views bereitzustellen	41

Controller	41
Benutzereingaben	44
Abhängigkeiten in Controller einfügen.	46
Ressourcen-Controller	47
API-Ressourcen-Controller	48
Controller für eine einzelne Aktion	48
Routen-Modell-Bindung	49
Implizite Routen-Modell-Bindung	49
Benutzerdefinierte Routen-Modell-Bindung	50
Routen-Caching	51
Methoden-Spoofing für Formulare	51
HTTP-Verben in Laravel	52
HTTP-Methoden-Spoofing	52
CSRF-Schutz	52
Umleitungen	54
redirect()->to()	55
redirect()->route()	55
redirect()->back()	56
Andere Umleitungsmethoden	56
redirect()->with()	57
Einen Request abbrechen	58
Gebräuchliche Response-Typen	59
response()->make()	59
response()->json() und ->jsonp()	59
response()->download(), ->streamDownload() und ->file()	59
Testen	60
TL;DR	61
4 Vorlagen erstellen mit Blade	63
Daten ausgeben	64
Kontrollstrukturen	65
Bedingungen	65
Schleifen	66
Vorlagen-Vererbung	67
Definieren von Abschnitten mit @section/@show und @yield. ...	68
Einbinden von Teilansichten	70
Verwendung von Stacks	72
Verwendung von Komponenten und Slots	73
View Composer und Service Injection	75
Daten mit View Composern an Views binden	76
Service Injection	78

Benutzerdefinierte Blade-Direktiven	79
Parameter in benutzerdefinierten Blade-Direktiven	81
Beispiel: Verwendung benutzerdefinierter Blade-Direktiven für eine mandantenfähige Anwendung	81
Einfachere benutzerdefinierte Direktiven für »if«-Anweisungen	82
Testen	83
TL;DR.	84
5 Datenbanken und Eloquent	85
Konfiguration	85
Datenbankverbindungen	86
Weitere Optionen zur Konfiguration von Datenbanken	87
Migrationen	88
Migrationen definieren.	88
Migrationen ausführen.	95
Seeding	96
Eine Seeder-Klasse anlegen	97
Modellfabriken.	98
Der Query Builder.	102
Grundlegender Einsatz der DB-Fassade.	103
Direktes SQL	104
Verkettung mit dem Query Builder	105
Transaktionen	114
Einführung in Eloquent	115
Erstellen und Definieren von Eloquent-Modellen	116
Abrufen von Daten mit Eloquent.	118
Inserts und Updates mit Eloquent	120
Löschen mit Eloquent	124
Geltungsbereiche	126
Anpassen von Feldinteraktionen durch Akzessoren, Mutatoren und Attribut-Casting	130
Eloquent-Collections	133
Serialisierung mit Eloquent	135
Beziehungen mit Eloquent	137
Aktualisierung von Zeitstempeln durch verknüpfte Datensätze	150
Ereignisse in Eloquent.	153
Testen	154
TL;DR.	156
6 Frontend-Komponenten	157
Laravel Mix	157
Verzeichnisstruktur von Mix	159

Mix ausführen.	159
Was bietet Mix?	160
Frontend-Frameworks und Auth-Scaffolding	167
laravel/ui	167
Frontend-Presets	168
Auth-Scaffolding	169
Paginierung	169
Paginieren von Datenbank-Ergebnissen	169
Paginator manuell erstellen	170
Message Bags	171
Benannte Error Bags	173
Hilfsfunktionen für Strings, Pluralisierung und Lokalisierung	173
Zeichenketten-Helfer und Pluralisierung	173
Lokalisierung	175
Testen	179
Message und Error Bags testen	179
Übersetzung und Lokalisierung	179
TL;DR	179
7 Benutzereingaben erfassen und verarbeiten	181
Injizieren eines Anforderungsobjekts	181
\$request->all()	182
\$request->except() und \$request->only()	182
\$request->has()	183
\$request->input()	183
\$request->method() und ->isMethod()	184
Benutzereingaben in Array-Form	184
JSON-Input (und \$request->json())	184
Routendaten	186
Daten aus dem Request-Objekt extrahieren	186
Daten aus Routenparametern	186
Hochgeladene Dateien	187
Validierung	189
validate() auf das Anforderungsobjekt anwenden	189
Manuelle Validierung	191
Benutzerdefinierte Regeln	192
Fehlermeldungen der Validierung anzeigen	193
Form Requests	194
Erstellen eines Form Requests	194
Verwendung eines Form Requests	195
Eloquent-Modelle und Massenzuweisung	196
{!! und {!!	197

Testen	198
TL;DR.	199
8 Artisan und Tinker	201
Eine Einführung in Artisan	201
Grundlegende Artisan-Befehle	202
Optionen	202
Befehle nach Gruppen	203
Benutzerdefinierte Artisan-Befehle	206
Ein Beispielbefehl	207
Argumente und Optionen	208
Benutzereingaben verwenden	210
Eingabeaufforderungen	212
Ausgaben	213
Schreiben von Closure-basierten Befehlen	214
Aufruf von Artisan-Befehlen in normalem Anwendungscode	214
Tinker	215
Laravels Dump-Server.	216
Testen	217
TL;DR.	218
9 Authentifizierung und Autorisierung	219
User-Modell und -Migration.	220
Verwendung des globalen auth()-Helfers und der Auth-Fassade	223
Die Auth-Controller	224
RegisterController	224
LoginController	225
ResetPasswordController.	227
ForgotPasswordController.	227
VerificationController	227
ConfirmPasswordController	227
Auth::routes()	228
Das Auth-Gerüst.	230
»Remember Me«: Die Erinnerungsfunktion	231
Manuelle Authentifizierung von Benutzern	233
Manuelles Abmelden eines Benutzers.	233
Invalidierung von Sitzungen auf anderen Geräten.	233
Auth-Middleware	234
E-Mail-Verifizierung	235
Blade-Direktiven zur Authentifizierung	236

Guards	236
Ändern des Standard-Wächters	237
Verwendung anderer Guards ohne Änderung des Standards	237
Hinzufügen eines neuen Guards	237
Closure Request Guards	238
Erstellen eines benutzerdefinierter Providers	238
Benutzerdefinierte Provider für nicht-relationale Datenbanken	239
Authentifizierungs-Ereignisse	239
Autorisierung (ACL) und Rollen	240
Berechtigungsregeln definieren	241
Die Gate-Fassade (und wie man Gate injiziert)	242
Gate für Ressourcen	243
Die Authorize-Middleware	243
Autorisierung per Controller	244
Überprüfen einer Instanz des User-Modells	245
Überprüfungen mit Blade	246
Abfangen von Prüfungen	246
Richtlinien	247
Testen	249
TL;DR	252
10 Request, Response und Middleware	253
Der Lebenszyklus des Request-Objekts	253
Bootstrapping der Anwendung	254
Service Provider	255
Das Request-Objekt	256
Zugriff auf das Request-Objekt in Laravel	257
Informationen aus einem Request erhalten	258
Das Response-Objekt	262
Response-Objekte in Controllern erzeugen und verwenden	262
Spezialisierte Antworttypen	263
Laravel und Middleware	269
Eine Einführung in Middleware	269
Benutzerdefinierte Middleware erstellen	270
Middleware binden	272
Parameter an die Middleware übergeben	275
Vertrauenswürdige Proxys	276
Testen	277
TL;DR	278

11 Der Container	279
Eine kurze Einführung in die Injektion von Abhängigkeiten	279
Abhängigkeitsinjektion und Laravel	281
Der globale Helfer app()	282
Wie ist der Container verdrahtet?	283
Klassen an den Container binden	284
Bindung mittels Closure.	284
Bindung von Singletons, Aliasen und Instanzen	285
Binden einer konkreten Instanz an ein Interface	286
Kontextuelle Bindung	287
Konstruktor-Injektion in Laravel-Framework-Dateien	287
Methoden-Injektion	288
Fassaden und Container	289
Wie Fassaden funktionieren	290
Echtzeit-Fassaden	291
Service Provider	292
Testen	292
TL;DR	293
12 Testen	295
Grundlagen des Testens	296
Tests benennen	300
Die Testumgebung	301
Vier spezielle Traits beim Testen	302
RefreshDatabase	302
WithoutMiddleware	302
DatabaseMigrations	302
DatabaseTransactions	303
Einfache Unit-Tests	303
Anwendungstests: So funktionieren sie	304
Die TestCase-Klasse	304
HTTP-Tests	305
Testen von Standardseiten mit \$this->get() und anderen HTTP-Aufrufen	305
Testen von JSON-APIs mit \$this->getJson() und anderen JSON-HTTP-Aufrufen	306
Behauptungen bezüglich \$response	307
Authentifizierung von Antworten	309
Weitere Anpassungen für HTTP-Tests	310
Behandlung von Ausnahmen in Anwendungstests	310

Datenbank-Tests	311
Verwendung von Modellfabriken	312
Seeding in Tests	312
Testen anderer Laravel-Features	312
Ereignisse faken	312
Bus- und Warteschlangen-Fakes	314
Mails faken	315
Benachrichtigungen faken	316
Dateioperationen faken	317
Mocking	317
Eine kurze Einführung ins Mocken	318
Eine kurze Einführung in Mockery	318
Andere Fassaden faken	321
Artisan-Befehle testen	322
Behauptungen bezüglich der Artisan-Befehlssyntax	322
Browser-Tests	323
Auswahl des Werkzeugs	323
Testen mit Dusk	324
TL;DR	335
13 APIs schreiben	337
Die Grundlagen REST-ähnlicher JSON-APIs	337
Controller-Organisation und JSON-Antworten	339
Header lesen und senden	342
Response-Header senden	343
Request-Header lesen	343
Paginierung	343
Sortieren und Filtern	345
Sortieren der API-Ergebnisse	345
Filtern der API-Ergebnisse	347
Ergebnisse transformieren	348
Schreiben eines eigenen Transformators	348
Verschachtelung und Beziehungen mit benutzerdefinierten Transformatoren	349
API-Ressourcen	351
Erstellen einer Ressourcen-Klasse	352
Ressourcen-Collections	353
Verschachtelte Beziehungen	354
Paginierung in API-Ressourcen verwenden	355
Bedingtes Anwenden von Attributen	356
Weitere Anpassungen für API-Ressourcen	356

API-Authentifizierung mit Laravel Passport	357
Eine kurze Einführung in OAuth 2.0	357
Passport installieren	357
Die Passport-API	359
Passports Grant-Typen	360
Clients und Tokens mit der Passport-API und Vue-Komponenten verwalten	368
Passport-Scopes	370
Bereitstellen von Passport	372
API-Token-Authentifizierung	373
Benutzerdefinierte 404-Antworten	374
Triggern der Fallback-Route	374
Testen	374
Passport testen	375
TL;DR.	375
14 Daten speichern und abrufen	377
Lokale und Cloud-basierte Datei-Manager.	377
Konfiguration des Dateizugriffs	377
Verwendung der Storage-Fassade	378
Zusätzliche Flysystem-Provider hinzufügen.	380
Grundlagen von Datei-Uploads und -Handhabung	380
Einfache Datei-Downloads	381
Sessions.	382
Zugriff auf die Session	382
Methoden, die für Session-Instanzen verfügbar sind	383
Flash-Sitzungsspeicher	385
Cache	385
Zugriff auf den Cache	386
Methoden, die für Cache-Instanzen verfügbar sind	386
Cookies	388
Cookies in Laravel	388
Auf Cookies zugreifen	388
Logging	391
Wann und warum man Logs verwenden sollte	392
In Logs schreiben	392
Log-Kanäle	393
Volltextsuche mit Laravel Scout	396
Scout installieren	396
Ein Modell für die Indexierung kennzeichnen	396
Einen Index durchsuchen	397
Warteschlangen und Scout	397

Operationen ohne Indizierung durchführen.	397
Bedingt ausgeführte Indizierung.	398
Manuelles Auslösen der Indizierung im Code	398
Manuelles Auslösen der Indizierung über die Kommandozeile	398
Testen	398
Dateien speichern	399
Session	400
Cache	401
Cookies	401
Logging	403
Scout	403
TL;DR	403
15 E-Mail und Benachrichtigungen	405
E-Mail	405
»Klassische« E-Mail	406
E-Mails als »Mailable«	406
E-Mail-Vorlagen	408
In build() verfügbare Methoden	409
Anhänge und Inline-Bilder	410
Markdown-Mailables	411
Darstellung von Mailables im Browser	413
Warteschlangen	413
Lokale Entwicklung	414
Benachrichtigungen	415
Definieren der via()-Methode für die zu benachrichtigenden	
Empfänger.	418
Senden von Benachrichtigungen.	419
Benachrichtigungen in Warteschlangen stellen	419
Laravels integrierte Benachrichtigungstypen	420
Testen	424
E-Mail	424
Benachrichtigungen	424
TL;DR	425
16 Queues, Jobs, Events, Broadcasting und der Scheduler	427
Warteschlangen	427
Warum Warteschlangen?	428
Grundlegende Warteschlangen-Konfiguration.	428
Warteschlangen-Jobs	429
Ausführen eines Queue-Workers	432
Fehlerbehandlung.	433

Kontrolle der Warteschlange	435
Warteschlange für andere Funktionen.	436
Laravel Horizon	436
Ereignisse	437
Ein Ereignis auslösen	438
Nach einem Ereignis »lauschen«	439
Broadcasting von Ereignissen über WebSockets und Laravel Echo	442
Konfiguration und Einrichtung	443
Übertragung eines Ereignisses	444
Empfang der Nachricht	446
Fortgeschrittene Broadcasting-Werkzeuge	449
Laravel Echo (die JavaScript-Seite)	452
Scheduler	457
Verfügbare Aufgabentypen	457
Verfügbare Zeitangaben.	458
Definieren von Zeitzonen für geplante Befehle	460
Blockierung und Überlappung.	460
Output von Aufgaben handhaben	460
Aufgaben-Hooks	461
Testen	462
TL;DR.	463
17 Helfer und Collections	465
Helfer	465
Arrays.	465
Zeichenketten.	467
Anwendungspfade	469
URLs	470
Verschiedenes.	472
Collections	475
Die Grundlagen	475
Ein paar Methoden	477
TL;DR.	482
18 Das Laravel-Ökosystem	483
Tools, die in diesem Buch behandelt werden	483
Valet	483
Homestead.	483
Der Laravel-Installer.	484
Mix.	484
Dusk.	484
Passport	484

Horizon.....	484
Echo	485
Tools, die in diesem Buch nicht behandelt werden.....	485
Forge.....	485
Envoyer.....	485
Cashier	486
Socialite.....	487
Nova	487
Spark.....	487
Lumen.....	487
Envoy	488
Telescope	488
Vapor	488
Weitere Ressourcen.....	489
Glossar.....	491
Index.....	499