

Auf einen Blick

TEIL I Einführung

1	Über dieses Buch	23
2	Installation und die Wahl der IDE	27
3	Warum Sie mit Dart voll ins Schwarze treffen	39
4	Pubs: Abhängigkeiten komfortabel verwalten	159

TEIL II Bedienoberflächen

5	Widgets	173
6	Layouting	207
7	Navigation	295
8	Früchte am Baum: InheritedWidgets	331
9	BloCs: View Logic aus dem Weg räumen	337
10	Mit dem Nutzer interagieren	359
11	Themes und Styles	395
12	Animationen	411
13	Arbeiten mit Canvas und Paint	429

TEIL III Assets und Medien

14	Assets und Medien	441
15	Den Nutzerkreis vergrößern: Localization und Accessibility	457
16	Die Cloud: Unendliche Weiten	487
17	Persistieren: Caches und Datenbanken	533

TEIL IV Sensoren und Hardware

18	Mit der Außenwelt kommunizieren	571
19	Den Kanal wechseln: Channels zur Plattform	581

TEIL V Endgame

20	Analytics, Tests und Releases	603
----	-------------------------------------	-----

Inhalt

Materialien zum Buch	19
----------------------------	----

TEIL I Einführung

1 Über dieses Buch	23
---------------------------	-----------

1.1 Was Sie in diesem Buch lernen werden	24
1.2 Null-Safety als Default-Einstellung ab Flutter 2.2	26
1.3 Danksagung	26

2 Installation und die Wahl der IDE	27
--	-----------

2.1 Flutter herunterladen und installieren	27
2.2 Die Wahl der IDE	31
2.2.1 IntelliJ und Android Studio	31
2.2.2 Visual Studio Code	34
2.3 Die Flutter-Installation voll im Griff	36

3 Warum Sie mit Dart voll ins Schwarze treffen	39
---	-----------

3.1 Der Ursprung von Dart	40
3.1.1 Die Köpfe hinter der Sprache	40
3.1.2 Die Kernideen	40
3.1.3 Eine Sprache des 21. Jahrhunderts	41
3.2 Ein erstes Dart-Programm	41
3.2.1 Dateinamen	43
3.2.2 Kommentare	43
3.3 Wichtige Begleiter	43
3.4 Variablen	44
3.4.1 Wiederholungstäter: Variablen mit var	44

3.4.2	Konstanten mit final und const	45
3.4.3	Warum Typinferenz?	46
3.5	Datentypen	48
3.5.1	Numerische Datentypen	49
3.5.2	Boolesche Werte	51
3.5.3	Strings	52
3.5.4	Datum, Zeit und Dauer	61
3.5.5	dynamic	63
3.5.6	Den Typ identifizieren	65
3.6	Funktionen	66
3.6.1	Der Bezeichner	67
3.6.2	Parameter	67
3.6.3	Der Funktionsrumpf	70
3.6.4	Synchrone und asynchrone Ausführung	72
3.6.5	Der Rückgabetyp	72
3.6.6	Der Typ Function	74
3.6.7	Anonyme Funktionen	74
3.6.8	Getter und Setter	76
3.7	Klassen, Mixins und Enums	77
3.7.1	Klassen	77
3.7.2	Object: Der Bottom-Type	86
3.7.3	Vererbung und Interfaces	89
3.7.4	Statische Elemente	96
3.7.5	Mixins	97
3.7.6	enums	99
3.7.7	typedef	101
3.7.8	Extension Methods	103
3.8	Generika	108
3.8.1	Generische Klassen	108
3.8.2	Generische Funktionen	109
3.8.3	Den Typparameter einschränken	110
3.8.4	Mehrere Typparameter	111
3.9	Collections	111
3.9.1	List	111
3.9.2	Set	117
3.9.3	Map	119
3.9.4	Iterable	122
3.10	Kontrollstrukturen	127
3.10.1	if-else	127

3.10.2	Die for-Schleife	130
3.10.3	for-in	131
3.10.4	while und do-while	131
3.10.5	Die Schlüsselwörter continue und break	132
3.10.6	switch-case	132
3.10.7	Assert	133
3.11	Exceptions und Errors	134
3.11.1	Exceptions	134
3.11.2	Errors	135
3.11.3	Fehler werfen: throw	136
3.11.4	Der zweite Parameter von catch	137
3.11.5	Spezifische Exceptions oder Errors fangen	137
3.11.6	Das Schlüsselwort finally	137
3.12	Asynchrone Programmierung	138
3.12.1	Event Loops	139
3.12.2	Future	139
3.12.3	async und await	143
3.12.4	Streams	145
3.12.5	Isolates	146
3.13	Die Library	147
3.13.1	Mini-Library	147
3.13.2	Eine Library importieren	147
3.14	Ein Projekt strukturieren	149
3.15	Code dokumentieren	150
3.16	Null-Safety	152
3.16.1	Nullability als Opt-In	152
3.16.2	Ein Projekt migrieren	153
3.17	Zusammenfassung	156
4	Pubs: Abhängigkeiten komfortabel verwalten	159
4.1	Von der Library zum Package	159
4.2	Ein Meer aus pubs	160
4.3	Wo Licht ist, da fällt auch Schatten	160
4.4	Packages verwenden	162

4.4.1	pubspec.yaml	163
4.4.2	Ein Package integrieren	163
4.5	Versionen	166
4.5.1	Caret-Syntax	166
4.5.2	Versionsrahmen	166
4.5.3	Feste Versionen	167
4.5.4	Keine Version angeben	168
4.5.5	Packages aktualisieren	168
4.5.6	Versionen erzwingen	168
4.6	Der System-Cache	169
4.7	Der Build Runner	169
4.8	Zusammenfassung	170

TEIL II Bedienoberflächen

5 Widgets	173
--------------------	------------

5.1	Ein erstes Widget	174
5.1.1	Stateless Widgets	182
5.2	Stateful Widgets	182
5.2.1	Die Ausgangssituation	183
5.2.2	Ein neues (Stateful)Widget erstellen	184
5.2.3	Eingabefelder layouten	186
5.2.4	Das Widget verschieben	187
5.2.5	Das State-Objekt	188
5.2.6	Die LoginMask mit State-Objekten anreichern	191
5.3	Material Design und Cupertino Design	194
5.3.1	Plattformübergreifende Widgets	196
5.4	Vom Widget zum Rendering Object	197
5.4.1	Widget-Tree	198
5.4.2	Element-Tree	198
5.4.3	RenderObject-Tree	200
5.5	Keys	201
5.5.1	Widgets referenzieren	201
5.5.2	Widgets markieren	203
5.6	Zusammenfassung	205

6	Layouting	207
6.1	Positionierung	207
6.1.1	Constraints	208
6.1.2	Zentrieren	208
6.1.3	Align	209
6.1.4	Padding	210
6.1.5	Baseline	211
6.1.6	SafeArea	212
6.1.7	Column	212
6.1.8	Row	216
6.1.9	Overflows mit Expanded und Flexible bekämpfen	217
6.1.10	Spacer	221
6.1.11	Tabellen	222
6.1.12	Stacks	225
6.1.13	Wrap	229
6.2	Scroll-Widgets	231
6.2.1	ListView	231
6.2.2	GridView	236
6.2.3	Der Konstruktor für Spezialfälle	241
6.2.4	Vertikal oder horizontal: ScrollDirection	241
6.2.5	Der ScrollController	242
6.2.6	Die Eigenschaft shrinkWrap	244
6.2.7	Ein einfaches Widget scrollen	246
6.2.8	Die Scrollleiste	247
6.2.9	ScrollNotifications	247
6.2.10	Wie im Karussell: Die PageView	249
6.2.11	Slivers	251
6.2.12	Caching von Items	256
6.3	Constraints, Höhen und Breiten	259
6.3.1	AspectRatio	259
6.3.2	ConstrainedBox	259
6.3.3	UnconstrainedBox	260
6.3.4	SizedBox	260
6.3.5	FittedBox	261
6.3.6	FractionallySizedBox	262
6.3.7	LimitedBox	262
6.3.8	OverflowBox	263
6.3.9	SizedOverflowBox	264

Inhalt

6.3.10	IntrinsicHeight und IntrinsicWeight	265
6.3.11	LayoutBuilder und SliverLayoutBuilder	266
6.4	MediaQueryData	267
6.4.1	Drei verschiedene Arten von Padding	269
6.4.2	Ein eigenes MediaQueryData-Objekt einfügen	272
6.4.3	Bedienungshilfen	273
6.5	Overlay	273
6.5.1	OverlayEntry	273
6.5.2	Overlay auflösen	274
6.5.3	OverlayEntry auf dem Overlay positionieren	275
6.5.4	Mit Overlay einen Hinweis anzeigen	275
6.6	Die Gerüste	283
6.6.1	Das Scaffold	283
6.6.2	CupertinoTabScaffold	288
6.6.3	CupertinoPageScaffold	289
6.6.4	Slivers in Scaffolds	290
6.6.5	Automatisches Scrollen	293
6.7	Zusammenfassung	294
7	Navigation	295
7.1	Ein erster Sprung	295
7.2	Wenn Flutter nach dem Weg fragt: Named Routes	299
7.2.1	InitialRoute und Routes	299
7.2.2	PushNamed und Pop	300
7.3	Eine Route austauschen	301
7.3.1	pushReplacementNamed und pushReplacement	301
7.3.2	popAndPushNamed	302
7.3.3	replace	303
7.4	Routen entfernen	303
7.5	Großflächig aufräumen mit push und pop	304
7.6	NavigatorObserver	304
7.7	Daten austauschen	305
7.7.1	Daten über den Konstruktor	306
7.7.2	Daten über RouteSettings	306
7.7.3	Wenn Routen antworten	307

7.8	Den Android-Back-Button zähmen	308
7.9	Routen als URIs	309
7.10	Verschachtelte Navigator-Objekte	309
7.11	Navigator 2.0	311
7.11.1	Wann brauchen Sie den Navigator 2.0?	311
7.11.2	Die Bestandsaufnahme	312
7.11.3	Von der RouteInformation zum Navigator	315
7.11.4	Die Startseite anzeigen	317
7.11.5	Die Navigation History ersetzen	322
7.11.6	Navigationsziele im Router mit onGenerateRoute	327
7.11.7	Zurücknavigation durch Plattformereignisse	328
7.12	Zusammenfassung	329

8 Früchte am Baum: InheritedWidgets

8.1	Ein InheritedWidget	331
8.2	Das eigene InheritedWidget: TimeInfo	332
8.3	Zusammenfassung	335

9 BloCs: View Logic aus dem Weg räumen

9.1	Der BloC	337
9.2	BloC und ValueListenable	338
9.3	ChangeNotifier	342
9.4	Streams	345
9.4.1	StreamController	345
9.4.2	Broadcasts	349
9.4.3	Fehler in Streams	351
9.5	Futures	352
9.6	Frische Pizza mit flutter_bloc	355
9.7	Zusammenfassung	358

10 Mit dem Nutzer interagieren 359

- 10.1 Button, TextField und Co** 359
 - 10.1.1 TextFields 362
 - 10.1.2 Den Fokus steuern 371
 - 10.1.3 GestureDetector 373
 - 10.1.4 Buttons 375
 - 10.1.5 Checkbox und Switch 377
- 10.2 Wischen, Schieben und Ziehen** 380
 - 10.2.1 Dismissible 380
 - 10.2.2 Draggable 382
 - 10.2.3 Slider 385
- 10.3 Dialogs und die Snackbar** 387
 - 10.3.1 Dialogs 388
 - 10.3.2 Snackbar 391

11 Themes und Styles 395

- 11.1 BoxDecoration** 395
 - 11.1.1 Ein Kreis mit Container 396
 - 11.1.2 Spezifische Radien mit Container 397
 - 11.1.3 Gradient in einer BoxDecoration 398
 - 11.1.4 Schatten 398
- 11.2 Clipping** 399
 - 11.2.1 ClipRect und ClipPath 399
 - 11.2.2 ClipRRect 401
 - 11.2.3 ClipOval 402
 - 11.2.4 ClipBehavior 402
- 11.3 Text** 403
 - 11.3.1 TextStyle 403
 - 11.3.2 RichText 404
 - 11.3.3 DefaultTextStyle 405
- 11.4 ThemeData und CupertinoThemeData** 406

	Inhalt
11.5 Effekte durch Widgets	406
11.5.1 Rotation	407
11.5.2 Transform	407
11.5.3 Opacity	409
11.5.4 BackdropFilter	409
 12 Animationen	 411
12.1 Implizite Animationen	411
12.1.1 Die Animated-Widgets	411
12.1.2 Überflieger: Hero in der Navigation	414
12.2 Tween-Animationen	416
12.2.1 TweenAnimationBuilder	416
12.2.2 Ein eigenes Tweenable	417
12.3 Explizite Animationen	419
12.3.1 AnimatedIcon	419
12.3.2 Transitions	421
12.4 AnimatedWidget und der AnimatedBuilder	424
12.4.1 AnimatedWidget	424
12.4.2 AnimatedBuilder	426
12.4.3 Widget oder Builder?	427
 13 Arbeiten mit Canvas und Paint	 429
13.1 CustomPaint und CustomPainter	429
13.1.1 CustomPaint integrieren	429
13.1.2 Formen und Farbe	430
13.1.3 Bilder	434
13.1.4 Text	434
13.1.5 Clipping	436
13.1.6 Weitere Operationen	436
13.2 Zusammenfassung	437

TEIL III Assets und Medien

14 Assets und Medien 441

14.1 Bilder	441
14.1.1 Bilder als Assets einbinden	442
14.1.2 Ein Image über eine URL anzeigen	443
14.1.3 Wenn ein Bild im Speicher liegt	443
14.1.4 Ein Bild vom Dateisystem auslesen	443
14.1.5 Caching	443
14.1.6 Eigenschaften von Image	444
14.1.7 SVGs anzeigen	446
14.2 Icons	446
14.2.1 IconData	447
14.2.2 IconButton	449
14.2.3 ImageIcon	449
14.2.4 IconTheme	450
14.3 Videos	450
14.4 Schriften	452
14.4.1 Eine Schrift hinzufügen	452
14.4.2 Eine Schrift benutzen	453
14.5 Textressourcen	453
14.6 Zusammenfassung	455

15 Den Nutzerkreis vergrößern: Localization und Accessibility 457

15.1 Localization	457
15.1.1 Die Lokalisierung in das Projekt einbinden	458
15.1.2 Eigene Texte und Übersetzungen	459
15.1.3 Datum und Zahlen	463
15.1.4 Intl in die IDE integrieren	467
15.1.5 Im Betrieb die Sprache ändern	472
15.1.6 Werte an die lokalisierten Texte übergeben	474

15.2	Accessibility	475
15.2.1	Schriftgröße	475
15.2.2	Screenreader	477
15.3	Zusammenfassung	486
 16 Die Cloud: Unendliche Weiten		487
16.1	Daten mit GET beziehen	488
16.2	Ein eigener (HTTP-)Client	489
16.2.1	Ein AppRequest	489
16.2.2	AppResponse	490
16.2.3	Der AppHttpClient	491
16.2.4	Die Methode get	491
16.3	POST, PUT und PATCH	493
16.3.1	http.post	493
16.3.2	POST im AppHttpClient	494
16.3.3	http.put und http.patch	497
16.3.4	PUT und PATCH im AppHttpClient	498
16.4	Delete	500
16.5	Mehr zu Requests	502
16.5.1	Header	502
16.5.2	Body	503
16.5.3	Query Parameters	503
16.5.4	Einen Proxy in Verbindung mit einem Client einsetzen	504
16.6	Das Package dio	505
16.6.1	Integration und erste Schritte	505
16.6.2	dio.get	506
16.6.3	BaseOptions	508
16.6.4	Query Parameters in dio	508
16.6.5	dio.post	509
16.6.6	Options	509
16.6.7	Das CancelToken	510
16.6.8	Den Fortschritt eines Requests anzeigen	510
16.6.9	Interceptors	511

16.7	JSON-Serialisierung und -Deserialisierung	512
16.7.1	json_serializable integrieren	513
16.7.2	Die Klasse TodoDto	514
16.7.3	Den build_runner hinzufügen	515
16.7.4	fromJson und toJson	515
16.7.5	TodoDto.fromJson einsetzen	516
16.8	Die Webview	517
16.8.1	Die Standard-Webview	518
16.8.2	Die InAppWebView	528
16.9	Zusammenfassung	532
17	Persistieren: Caches und Datenbanken	533
17.1	shared_preferences	534
17.1.1	Das Package integrieren	534
17.1.2	In den Cache schreiben	534
17.1.3	Aus dem Cache lesen	535
17.1.4	Weitere nützliche Methoden	537
17.1.5	Einträge entfernen	538
17.2	flutter_cache_manager	538
17.2.1	Integration und Konfiguration	539
17.2.2	Dateien herunterladen	539
17.2.3	Ablegen, löschen und das direkte Lesen aus dem Cache	541
17.3	Datenbanken	542
17.3.1	Die Basis: sqflite	543
17.3.2	Mehr Abstraktion durch Drift	548
17.4	NoSQL mit Hive CE	554
17.5	Eine wirklich nützliche Beispiel-App	556
17.5.1	Ein Mini-Projektplan	556
17.5.2	Die Datenklasse	557
17.5.3	Die Datenbank-Klasse	559
17.5.4	Die Main-Funktion	562
17.5.5	Die UI	563
17.6	Zusammenfassung	567

TEIL IV Sensoren und Hardware

18 Mit der Außenwelt kommunizieren 571

18.1	Geräteinformationen	571
18.2	Bewegungssensoren und Batterie	573
18.3	Verbindungsdaten	574
18.4	Berechtigungen	575
18.5	Kamera und Fotobibliothek	576
18.6	Standortbestimmung	577
18.7	Zusammenfassung	579

19 Den Kanal wechseln: Channels zur Plattform 581

19.1	Method Channels	582
19.1.1	Ein Flutter-Plugin erstellen	583
19.1.2	Die Anatomie eines Plugins	583
19.1.3	Zusammenfassung	587
19.2	Foreign Function Interface (FFI)	588
19.2.1	Überblick	588
19.2.2	Die »C-Library«	588
19.2.3	Verwendung der Schnittstellen in Dart	590
19.3	Hybride Apps	592
19.3.1	Native Views in Flutter	592
19.3.2	Eine Android-View einbinden	592
19.3.3	Eine iOS-View einbinden	596

TEIL V Endgame

20 Analytics, Tests und Releases 603

20.1	Logging	603
20.2	Analytics	605

- 20.3 Debugging** 605
 - 20.3.1 Hot-Reload, Hot-Restart und Full Restart 605
 - 20.3.2 Flutter Inspector 606
 - 20.3.3 Breakpoints 607
- 20.4 Profiling** 607
- 20.5 Testing** 609
 - 20.5.1 Unit-Testing 609
 - 20.5.2 Widget-Testing 617
- 20.6 Releases** 620
 - 20.6.1 Ein eigenes Pub erstellen 620
 - 20.6.2 Android-Releases 623
 - 20.6.3 iOS-Releases 623
 - 20.6.4 Native und Web-Bundles 624
- 20.7 Zusammenfassung** 624

- Index 625