

Inhalt

Einleitung	13
------------------	----

TEIL I No-Code-Entwicklung mit SAP Build Apps

1 Überblick 21

1.1 Was bedeutet Low Code/No Code?	22
1.2 Welche Vorteile bringt Low Code/No Code im SAP-Umfeld?	30
1.3 Was ist SAP Build Apps?	32
1.3.1 Benutzeroberflächen mittels Drag & Drop erstellen	34
1.3.2 Logik visuell erstellen	35
1.3.3 Datenintegration	36
1.3.4 Formelfunktionen nutzen	37
1.3.5 Anwendungen veröffentlichen	38
1.3.6 Theme einstellen	38
1.3.7 Marketplace	39
1.4 Was ist SAP Build?	40
1.5 Was ist SAP Fiori?	41
1.5.1 SAP Fiori als User Experience	42
1.5.2 SAP Fiori Elements	45

2 Einführung in die Entwickertools von SAP Build Apps 51

2.1 Einführung in die Community Edition (ehemals Composer Pro)	52
2.2 Einführung in den Service SAP Build Apps auf der SAP BTP	70

3 Ihre erste Business-App 87

3.1 Projekt anlegen	88
3.2 Darstellungselemente hinzufügen	91
3.3 App testen	98

4 Für Admins: SAP Build Apps einrichten 105

4.1 SAP Build Apps im SAP-BTP-Subaccount aktivieren	106
4.1.1 Anlegen eines Subaccounts	107
4.1.2 Zuweisen der Entitlements	109
4.1.3 Option 1: Manuelle Aktivierung	112
4.1.4 Option 2: Verwenden eines SAP-BTP-Boosters	118
4.2 Möglichkeiten der Authentifizierung und Autorisierung	122
4.3 Lifecycle Management in SAP Build Apps	127
4.4 Anlegen von Destinations zur Verwendung von OData-Services	128

TEIL II SAP-Anwendungen mit SAP Build Apps entwickeln

5 Wie kommen die Daten in die App? 133

5.1 Datenmodellierung	134
5.2 Serviceanbindung	139
5.3 Grundlagen von REST-APIs	143
5.4 Grundlagen von OData-APIs	159
5.4.1 Classic Data Entity in SAP Build Apps anlegen	160
5.4.2 OData-Service mit einer SAP-BTP-Destination integrieren	166

6 Was soll Ihre App können? 173

6.1	Eine Seite erstellen	174
6.2	Inhalte anzeigen	180
6.2.1	Statische Inhalte	180
6.2.2	Dynamische Inhalte	182
6.3	Variablen verwenden	192
6.3.1	Arten von Variablen	193
6.3.2	Eine Variable in der App verwenden	201
6.4	Seiten und Links aufrufen	206
6.5	Auf Daten zugreifen und speichern	213
6.6	Auf Events reagieren	220

7 Wie soll Ihre App aussehen? 223

7.1	Grundlagen zu Benutzeroberflächen	224
7.2	Verfügbare UI-Komponenten	227
7.3	Formulare erstellen	253
7.4	Daten in Listen ausgeben	258
7.5	Layouts erstellen	268

8 Tiefer in die App-Entwicklung einsteigen 275

8.1	Eigenen JavaScript-Code einfügen	275
8.1.1	JavaScript mit einem Eingangs- und einem Ausgangsparameter	276
8.1.2	JavaScript mit mehreren Eingangs- und Ausgangsparametern	283
8.2	App an Ihr Corporate Design anpassen	290
8.3	Custom Controls erstellen	297

9 App veröffentlichen

305

9.1 App auf der SAP BTP zur Verfügung stellen	307
9.1.1 App als ZIP-Datei auf der SAP BTP bereitstellen	311
9.1.2 App als MTAR-Datei auf der SAP BTP bereitstellen	317
9.2 Mobile Apps zur Verfügung stellen	320
9.2.1 Apps für das iOS-Betriebssystem anbieten	321
9.2.2 Apps über Android anbieten	324

TEIL III Entwicklungsbeispiele und Einsatzbereiche

10 Entwicklung einer App für Aufgabenlisten 331

10.1 App entwickeln	331
10.1.1 App-Struktur anlegen	332
10.1.2 Datenspeicherung konfigurieren	338
10.1.3 Darstellungslogik und Data Binding anlegen	340
10.1.4 Datenübergabe zwischen Seiten einrichten	343
10.1.5 Seitenladelogik anlegen	346
10.1.6 Datenspeicherung einstellen	352
10.2 App testen	357

11 Entwicklung einer GPS-Tracking-App 359

11.1 App entwickeln	360
11.2 App testen	367
11.3 Erweiterung einbauen	367

12 Entwicklung einer Kundenverwaltungs-App	381
12.1 App entwickeln	382
12.1.1 Benutzeroberfläche gestalten und Seiten anlegen	383
12.1.2 Verbindung zu Google Firebase einrichten	387
12.1.3 Daten aus einer Google-Firebase-Datenbank speichern und auslesen	394
12.2 App testen	410
12.3 App erweitern	412
13 Entwicklung einer App zur Anzeige von Produktinformationen	427
13.1 App entwickeln	427
13.2 App auf einem mobilen Endgerät testen	451
14 Entwicklung eines Formulars zur Datenausgabe und -pflege	455
14.1 Variante 1: OData-Service über eine SAP-BTP-Destination einbinden	456
14.2 Variante 2: OData-Service direkt einbinden	462
14.3 App entwickeln	465
14.4 App testen	487
Die Autoren	491
Index	493