

Auf einen Blick

TEIL I Konzepte des ABAP RESTful Application Programming Model

1	Einführung in das ABAP RESTful Application Programming Model	27
2	Core Data Services: Datenmodellierung	71
3	Verhaltensdefinition	133
4	Entity Manipulation Language: Zugriff auf Geschäftsobjekte	229
5	Verhaltensimplementierung	249
6	Business-Services	291
7	Anwendungsoberflächen und SAP Fiori Elements	307

TEIL II Praktische Anwendungsentwicklung mit dem ABAP RESTful Application Programming Model

8	Anwendungsfälle	349
9	Managed Scenario: Entwicklung einer Anwendung mit SAP Fiori Elements	357
10	Managed Scenario mit Unmanaged Save: Bestehende Anwendung integrieren	411
11	Unmanaged Scenario: Vorhandenen Quellcode wiederverwenden	473
12	Besonderheiten im SAP BTP, ABAP Environment	513
13	Ausblick	541

Inhalt

Vorwort	17
Einleitung	19

TEIL I Konzepte des ABAP RESTful Application Programming Model

1 Einführung in das ABAP RESTful Application Programming Model	27
1.1 Was ist das ABAP RESTful Application Programming Model? ...	28
1.1.1 Aufgaben des Programmiermodells	28
1.1.2 Der REST-Architekturstil	33
1.1.3 OData	39
1.1.4 Technologische Neuerungen mit SAP S/4HANA	43
1.1.5 Evolution der ABAP-basierten Programmiermodelle	44
1.2 Architektur und Konzepte des ABAP RESTful Application Programming Model	49
1.2.1 RAP-Transaktionsmodell	50
1.2.2 Implementierungstypen	51
1.2.3 Entity Manipulation Language	52
1.2.4 Technischer Kontext einer RAP-Anwendung und RAP-Laufzeitumgebung	53
1.3 Entwicklungsobjekte des ABAP RESTful Application Programming Model	56
1.3.1 Datenmodellierung mit Core Data Services	56
1.3.2 Verhaltensdefinition	57
1.3.3 Verhaltensimplementierung	58
1.3.4 Projektionsschicht	59
1.3.5 Business-Services	60
1.3.6 Zusammenspiel der Artefakte	60
1.4 ABAP Development Tools als Entwicklungswerkzeug	61
1.5 Qualitative Eigenschaften des ABAP RESTful Application Programming Model	63
1.5.1 Evolutionsfähigkeit	63

1.5.2	Entwicklungseffizienz	64
1.5.3	Testbarkeit	66
1.5.4	Trennung zwischen Fachlichkeit und Technik	66
1.6	Verfügbarkeit des ABAP RESTful Application Programming Model	67
1.6.1	SAP BTP, ABAP Environment	68
1.6.2	ABAP-Plattform für SAP S/4HANA on-premise	69

2 Core Data Services: Datenmodellierung

2.1	Was sind Core Data Services?	72
2.2	Aufbau und Syntax von Core Data Services	76
2.2.1	Erstellung eines Basic Interface View	77
2.2.2	Analyse des Datenmodells	81
2.2.3	CDS-Views verwenden	85
2.2.4	Erweiterung des Datenmodells	86
2.3	Assoziationen	91
2.4	Annotationen	96
2.5	Zugriffskontrollen	102
2.6	Erweiterbarkeit von CDS-Entitäten	108
2.6.1	CDS-View-Erweiterungen	108
2.6.2	CDS-Metadatenerweiterung	112
2.7	Weitere CDS-Funktionalität	115
2.7.1	Virtuelle Elemente	116
2.7.2	CDS Custom Entities	120
2.8	Virtuelles Datenmodell	124
2.9	CDS-Sprachelemente zur Modellierung von Geschäftssobjekten	129

3 Verhaltensdefinition

3.1	Was ist eine Verhaltensdefinition?	134
3.1.1	Kontext und Aufbau einer Verhaltensdefinition	134
3.1.2	Syntax einer Verhaltensdefinition	138
3.1.3	Mögliche Verhalten	139

3.2 Verhaltensdefinition in den ABAP Development Tools	
bearbeiten	146
3.2.1 Eine Verhaltensdefinition anlegen	146
3.2.2 Eine Verhaltensdefinition ändern und aktivieren	150
3.2.3 Eine Verhaltensdefinition suchen und öffnen	151
3.2.4 Verhaltensdefinitionen und Beziehungen dokumentieren	152
3.3 Implementierungstypen	154
3.3.1 Managed Scenario	156
3.3.2 Unmanaged Scenario	158
3.4 Strict-Modus	159
3.5 Entitätsverhaltensdefinition	160
3.6 Behavior Pool definieren	161
3.6.1 Behavior Pool zur Verhaltensdefinition	161
3.6.2 Behavior Pool zur CDS-Entität	162
3.6.3 Behavior Pool zur Implementierungsgruppe	163
3.7 Nummernvergabe	164
3.7.1 Frühe, externe Nummernvergabe	166
3.7.2 Frühe, interne Nummernvergabe	166
3.7.3 Späte Nummernvergabe	168
3.8 Feldeigenschaften	169
3.8.1 Pflichtfelder	170
3.8.2 Schutz vor schreibenden Zugriffen	170
3.8.3 Kombination: Pflichtfeld im Anlagefall, Schreibschutz bei Änderungen	172
3.9 Feld-Mappings	172
3.10 Standard-Operationen für eine CDS-Entität	175
3.10.1 Create, Read, Update und Delete	175
3.10.2 Create- und Read-Operation per Assoziation	176
3.11 Spezifische Operationen für eine CDS-Entität	179
3.11.1 Aktionen	179
3.11.2 Funktionen	187
3.12 Konkurrierende Zugriffe und Sperrverhalten	189
3.12.1 Pessimistisches Sperren	190
3.12.2 Optimistisches Sperren	192

3.13	Interne Geschäftslogik	195
3.13.1	Ermittlungen	195
3.13.2	Validierungen	201
3.13.3	Ermittlungen über Aktion aufrufen	204
3.14	Berechtigungsprüfungen	207
3.14.1	Authorization-Master	208
3.14.2	Authorization-Dependent	210
3.14.3	Berechtigungsprüfungen delegieren	212
3.15	Draft-Handling	213
3.15.1	Draft-Handling aktivieren	214
3.15.2	Draft-Handling im Geschäftsblobjekt-Kompositionbaum	214
3.15.3	Draft-Lebenszyklus und Draft-Aktionen	217
3.16	Übergreifende Konzepte	220
3.16.1	Dynamisches Feature Control	220
3.16.2	Vorprüfungen von Operationen	225
3.16.3	Interne Sichtbarkeit von Operationen	226

4 Entity Manipulation Language: Zugriff auf Geschäftsblobjekte

4.1	Datentypen	230
4.1.1	Abgeleitete Datentypen	231
4.1.2	Implizite Rückgabeparameter	233
4.2	EML-Operationen	234
4.2.1	READ ENTITIES	235
4.2.2	MODIFY ENTITIES	237
4.2.3	GET PERMISSIONS	241
4.2.4	SET LOCKS	242
4.2.5	COMMIT ENTITIES	243
4.2.6	ROLLBACK ENTITIES	244
4.3	Verwendung von EML außerhalb von Verhaltensimplementierungen	245
4.3.1	Verwendung im Rahmen eines ABAP Reports	245
4.3.2	Implementierung im Rahmen einer Testklasse	247

5 Verhaltensimplementierung

249

5.1 Business Object Provider API	249
5.2 Laufzeitverhalten des ABAP RESTful Application Programming Model	250
5.2.1 Interaktionsphase und Transaktionspuffer	251
5.2.2 Speichersequenz	253
5.3 Interfaces für den Interaktionshandler und den Speicherhandler	254
5.4 Interaktionshandler	255
5.4.1 FOR MODIFY	256
5.4.2 FOR INSTANCE AUTHORIZATION	260
5.4.3 FOR GLOBAL AUTHORIZATION	262
5.4.4 FOR FEATURES	265
5.4.5 FOR GLOBAL FEATURES	268
5.4.6 FOR LOCK	269
5.4.7 FOR READ	272
5.4.8 FOR READ per Assoziation	273
5.4.9 FOR DETERMINE	275
5.4.10 FOR VALIDATE	276
5.4.11 FOR NUMBERING	277
5.4.12 FOR PRECHECK	279
5.5 Speicherhandler	280
5.5.1 FINALIZE	281
5.5.2 CHECK_BEFORE_SAVE	283
5.5.3 ADJUST_NUMBERS	284
5.5.4 SAVE	286
5.5.5 CLEANUP	288
5.5.6 CLEANUP_FINALIZE	289

6 Business-Services

291

6.1 Projektionsschicht	292
6.1.1 CDS Projection View	293
6.1.2 Projektions-Verhaltensdefinition	294
6.2 Servicedefinition	296
6.3 Service-Binding	297

6.4	Business-Services im SAP Gateway Client testen	301
6.5	UI-Services mit der SAP-Fiori-Elements-Vorschau testen	304

7 Anwendungsoberflächen und SAP Fiori Elements

7.1	Entwicklungswerzeuge	308
7.1.1	SAP Business Application Studio	308
7.1.2	Visual Studio Code	311
7.2	SAP-Fiori-Elements-Oberflächen für RAP-Anwendungen	312
7.2.1	Floorplans in SAP Fiori Elements	312
7.2.2	Ausgewählte UI-Annotationen	315
7.2.3	UI-Annotationen in einem CDS-View definieren	316
7.2.4	Annotationen mit dem Service Modeler generieren	334

TEIL II Praktische Anwendungsentwicklung mit dem ABAP RESTful Application Programming Model

8 Anwendungsfälle

8.1	Einsatzgebiete des ABAP RESTful Application Programming Model	349
8.2	Abgrenzung der verschiedenen Implementierungstypen	350
8.3	Entscheidungskriterien zur Auswahl des Implementierungstyps	352

9 Managed Scenario: Entwicklung einer Anwendung mit SAP Fiori Elements

9.1	Beschreibung des Anwendungsfalls	358
9.2	Datenmodell aufbauen	358
9.2.1	Datenbanktabellen	359

9.2.2	CDS-Modellierung	362
9.3	Verhaltensdefinitionen erstellen	371
9.3.1	Verhaltensdefinitionen für die Zertifikatsverwaltung anlegen	371
9.3.2	Draft-Handling aktivieren	376
9.4	Business-Service definieren	378
9.4.1	Servicedefinition anlegen	378
9.4.2	Service-Binding erstellen	380
9.5	SAP-Fiori-Elements-Oberfläche erstellen	382
9.6	Anreicherung um eine Ermittlung	389
9.7	Anreicherung um eine Validierung	395
9.8	Anreicherung um eine Aktion	399
9.9	Generierung und Deployment der Anwendung	402

10 Managed Scenario mit Unmanaged Save: Bestehende Anwendung integrieren

10.1	Beschreibung des Anwendungsfalls	412
10.2	Datenmodell aufbauen	415
10.2.1	Überblick über das logische Datenmodell	416
10.2.2	Datenbanktabellen	418
10.2.3	CDS-Modellierung	420
10.3	Verhaltensdefinition erstellen	426
10.4	Funktion »Bestellung anlegen« realisieren	428
10.4.1	Späte Nummernvergabe deklarieren	428
10.4.2	Feldeigenschaften setzen	429
10.4.3	Behavior Pool anlegen	431
10.4.4	Ermittlungen realisieren	432
10.4.5	Speichersequenz: Anlage per BAPI realisieren	439
10.4.6	Validierungen realisieren	444
10.5	Funktion »Bestellung löschen« realisieren	451
10.5.1	Speichersequenz: Löschen per BAPI realisieren	451
10.5.2	Validierung realisieren	456

10.6 Business-Services definieren	458
10.6.1 Projektionsschicht für die Anwendung »Eigene Bestellungen« aufsetzen	458
10.6.2 Servicedefinition anlegen	461
10.6.3 Service-Binding anlegen	461
10.7 Berechtigungsprüfungen realisieren	462
10.7.1 Zugriffskontrollen für Lesezugriffe	462
10.7.2 Zugriffskontrollen für Schreibzugriffe	464
10.8 SAP-Fiori-Elements-Oberfläche erstellen	467
10.8.1 Metadatenerweiterung anlegen	467
10.8.2 Generierung und Deployment der Anwendung	470

11 Unmanaged Scenario: Vorhandenen Quellcode wiederverwenden

473

11.1 Beschreibung des Anwendungsfalls	474
11.2 Beschreibung der bestehenden Anwendung	475
11.2.1 Datenbanktabellen	475
11.2.2 Quellcode der bestehenden Anwendung	479
11.3 Datenmodell erweitern	482
11.4 Verhaltensdefinition erstellen	489
11.5 Verhaltensimplementierung erstellen	493
11.5.1 Implementierung der Interaktionsphase	496
11.5.2 Implementierung der Speichersequenz	504
11.6 Business-Service definieren	508

12 Besonderheiten im SAP BTP, ABAP Environment

513

12.1 Technische Grundlagen	514
12.1.1 ABAP for Cloud Development	517
12.1.2 Technische Infrastrukturkomponenten	518
12.1.3 Migration von Bestands-Coding	521
12.2 Identity and Access Management	522

12.3 SAP-Fiori-Apps bereitstellen und Berechtigungen vergeben	525
12.3.1 IAM-App und -Anwendungskatalog anlegen	527
12.3.2 IAM-Anwendungsrolle anlegen	529
12.3.3 Integration ins SAP Fiori Launchpad	530
12.4 Business-Services konsumieren	534

13 Ausblick

13.1 Build	542
13.2 Erweiterbarkeit	542
13.3 Integration und Wiederverwendbarkeit	544

Anhang

A Literaturverzeichnis	547
-------------------------------------	-----

Das Autorenteam	549
Index	551