

Karl Kessler, Peter Tillert, Panayot Dobrikov

Java-Programmierung mit dem SAP Web Application Server

Inhalt

Vorwort	13
1 SAP NetWeaver	17
1.1 Technologieplattform für Enterprise-Software	17
1.1.1 Motivation für die Plattform-Einführung	17
1.1.2 SAP-Basis: Eine erfolgreiche Technologieplattform	17
1.1.3 Architekturmerkmale der SAP-Basis	18
1.2 Integrationsplattform SAP NetWeaver	20
1.2.1 Integration innerhalb eines Systems	20
1.2.2 Integration durch Standards	21
1.2.3 Integrationsszenario Rechnungsprüfung	21
1.2.4 Komponenten-Architektur der Rechnungsprüfung	22
1.2.5 Anforderungen an eine Integrationsplattform	23
1.3 Komponenten von SAP NetWeaver	24
1.3.1 People und Information Integration	25
1.3.2 Process Integration	29
1.3.3 Application Platform	32
1.4 Ausblick: SAP NetWeaver und Enterprise Services Architecture	34
2 SAP NetWeaver Developer Studio: Features, Tools und Perspektiven	37
2.1 Benutzeroberfläche	37
2.2 Workspace, Projekte und Entwicklungsobjekte	40
2.3 Features	42
2.3.1 Integration der Java Development Infrastructure (JDI)	42
2.3.2 Server-Integration im Developer Studio	47
2.4 Architektur	53
2.5 Tools und Perspektiven	56
2.5.1 Development-Configuration-Perspektive	56
2.5.2 Dictionary-Perspektive	58
2.5.3 J2EE-Perspektive	61
2.5.4 Web-Dyn pro-Perspektive	64
2.5.5 Webservice-Perspektive	66
2.5.6 DTR-Perspektive	68

3 SAP NetWeaver Developer Studio: Step by Step zur Beispielanwendung 71

3.1	Employee-Beispielanwendung.....	72
3.2	Erste Schritte im SAP NetWeaver Developer Studio.....	73
3.2.1	Das Developer Studio starten.....	73
3.2.2	Einstellungen unter Windows-Preferences.....	74
3.3	Datenmodell definieren.....	75
3.3.1	Dictionary-Projekt anlegen.....	75
3.3.2	Employee-Tabelle definieren.....	77
3.3.3	Tabelle deployen.....	80
3.4	Zugriffe auf Tabellendaten realisieren.....	82
3.4.1	EJB-Modul-Projekt anlegen.....	82
3.4.2	Employee-Entity-Bean definieren.....	83
3.4.3	Data-Transfer-Objekt-Klasse erstellen.....	87
3.5	Business-Logik definieren.....	89
3.5.1	Session Bean EmployeeServices anlegen.....	89
3.5.2	Bean-Klasse implementieren.....	91
3.6	Deployment-Beschreibungen hinzufügen.....	94
3.6.1	Beschreibungen im ejb-jar.xml hinzufügen.....	95
3.6.2	Server-spezifische Deployment-Beschreibungen hinzufügen, . . .	100
3.6.3	Java-Archiv erstellen.....	102
3.7	JSP-basierte Web-Anwendung erstellen.....	102
3.7.1	Web-Modul-Projekt anlegen.....	103
3.7.2	Benutzeroberfläche mit JSP implementieren.....	104
3.7.3	Beschreibungen im Deployment-Deskriptor web.xml.....	107
3.7.4	Web-Archiv erstellen.....	109
3.8	J2EE-Gesamtanwendung definieren und deployen.....	110
3.8.1	Enterprise-Application-Projekt erstellen.....	110
3.8.2	Beschreibungen zu application.xml hinzufügen.....	110
3.8.3	Data-Source-Alias anlegen.....	111
3.8.4	Enterprise Application Archive {EAR} erstellen und deployen, . . .	112
3.9	Business-Logik als Webservice bereitstellen.....	113
3.9.1	Webservice anlegen.....	113
3.9.2	Webservice deployen.....	115
3.9.3	Webservice in der Testumgebung aufrufen.....	115

4 Java-Persistenz im SAP Web Application Server 117

4.1	Open JDBC für Java.....	117
4.1.1	Features von Open JDBC für Java.....	118
4.2	Persistenz-Infrastruktur des SAP Web AS zur Laufzeit.....	118
4.2.1	VendorJDBC.....	119

4.2.2	Native JDBC.....	119
4.2.3	Statement Pooling.....	120
4.2.4	SQLTrace.....	121
4.2.5	Tabellen-Pufferung.....	122
4.2.6	Administration von Datenbankverbindungen.....	122
4.3	Java Dictionary.....	123
4.3.1	Definition von Datenbanktabellen im SAP NetWeaver Developer Studio.....	124
4.4	Entwicklung einer Beispielanwendung.....	126
4.4.1	Beispiel-Szenario Projektmanagement.....	126
4.4.2	Definition und Deployment der Tabellen.....	128
4.4.3	Implementierung der Hilfsklassen.....	129
4.4.4	Implementierung der Session Bean und Deployment der Anwendung.....	130
4.5	Open SQL/SQU.....	131
4.5.1	Open SQL/SQU: Grundlegende Prinzipien.....	131
4.5.2	Implementierung des Projektmanagement-Szenarios mit Open SQL/SQU.....	132
4.6	Enterprise Java Beans - Container Managed Persistence.....	143
4.6.1	Implementierung des Projektmanagement-Szenarios mittels EJB CMP.....	144
4.7	Programmierung mit Java Data Objects (JDOJ).....	156
4.7.1	Konzepte von JDO.....	157
4.7.2	Vorbereitung des JDO-Projektes.....	158
4.7.3	Programmierung der Beispielklassen.....	159
4.7.4	Modifikation des Bytecodes mit dem Enhancer.....	162
4.7.5	Die objekt-relationale Abbildung persistenter Klassen auf die Datenbank.....	167
4.7.6	Programmierung der Anwendungslogik.....	170
4.7.7	Deployment der Anwendung.....	177
4.8	Persistenz für Experten.....	179

5 Webservices im SAP Web Application Server 181

5.1	Webservice Framework.....	182
5.2	Standardisierung von Webservices.....	184
5.3	Bereitstellung eines Webservice - die Server-Seite.....	186
5.3.1	Webservice anlegen.....	187
5.3.2	Webservice testen mit der Webservice-Homepage.....	196
5.3.3	Webservice sichern.....	198
5.3.4	Restriktionen für Webservice-Endpunkte.....	209
5.3.5	Unterstützte Typen in Webservice-Endpunkten.....	209
5.4	Webservices in öffentlichen Verzeichnissen anbieten.....	210
5.4.1	Webservice in einer Registry publizieren.....	218

5.5	Einbinden eines Webservice - die Client-Seite.....	219
5.5.1	Abrufen der WSDL-Beschreibung des Webservice.....	220
5.5.2	Generieren eines Webservice-Proxy.....	221
5.5.3	Implementieren der Client-Anwendung.....	223

6 Web Dynpro: Bedienoberflächen entwickeln 231

6.1	Arbeiten mit dem View Designer.....	231
6.1.1	Views und Layouts.....	231
6.1.2	Lokale Daten: Der Context eines Views.....	234
6.1.3	Daten binden.....	236
6.2	Interaktive Formulare mit Technologie von Adobe.....	239
6.2.1	Adobe Forms Designer im Developer Studio.....	239
6.2.2	Weiter gehende Szenarien mit interaktiven Formularen.....	241
6.3	Anwendungen mit mehreren Views.....	242
6.3.1	Datenübertragung zwischen mehreren Views durch Mapping ...	243
6.3.2	Views im Navigation Modeler anordnen.....	245
6.3.3	Roundtrips und Actions.....	246
6.3.4	Navigation von einem View zum anderen.....	248
6.4	Generierter und eigener Quellcode: Mit Event-Handleern arbeiten.....	251
6.4.1	Event-Handler für Actions.....	251
6.4.2	Andere Arten von Event-Handleern.....	253
6.4.3	API der Web-Dynpro-Laufzeitumgebung.....	254
6.4.4	Einfache Code-Beispiele für das Web-Dynpro-API.....	255
6.5	Dynamische Veränderungen der Bedienoberfläche.....	260
6.5.1	Context-Elemente dynamisch erzeugen.....	260
6.5.2	Bedienelemente dynamisch erzeugen.....	261
6.5.3	Bedienelemente an Actions binden.....	263
6.5.4	Statisch und dynamisch erzeugte Elemente.....	264
6.6	Web-Dynpro-Components.....	264
6.6.1	Components und Applications.....	265
6.6.2	Components als wieder verwendbare Einheiten.....	266
6.6.3	Beispiel für eine wieder verwendbare Component.....	268
6.7	Kommunikation zwischen Controllern.....	271
6.7.1	Verwendungsbeziehungen.....	272
6.7.2	Kommunikation über Events.....	273
6.8	Zusammenfassung.....	274
6.8.1	Meta-Modell von Web Dynpro.....	275
6.8.2	Ausblick.....	277

7 Web Dynpro: Geschäftsanwendungen entwickeln 279

7.1	Web Dynpro ruft Webservice - In fünf Schritten zur Anwendung.....	280
7.1.1	Importieren des Modells zum Employee-Webservice.....	283
7.1.2	Anbindung eines Custom Controllers an das Modell.....	289
7.1.3	Definition des Context-Mappings.....	293
7.1.4	View-Layout und Datenbindung.....	295
7.1.5	Controller-Implementierung.....	297
7.2	Web-Dynpro-Controller und ihre Schnittstellen.....	305
7.2.1	AAodel-View-Controller-Modell.....	306
7.2.2	Controller-Konzept.....	307
7.2.3	Generierte und generische Controller-APIs.....	313
7.2.4	Shortcut-Variablen wdControllerAPI und wdComponentAPI. . .	315
7.3	Contexte und Datenfluss zwischen Backend und Frontend.....	319
7.3.1	Context-Konzept.....	320
7.3.2	Typisierte und generische Context-APIs.....	322
7.3.3	Grundlagen der Context-Programmierung.....	325
7.3.4	Supply Functions.....	327
7.3.5	Kalkulierte Context-Attribute.....	331
7.3.6	Web-Dynpro-Models und Common Model Interface.....	333
7.3.7	Zugriff auf die Geschäftslogik mit dem Common Model Interface.....	335
7.3.8	Unterschiede zwischen Model- von Value-Knoten im Context.....	337
7.4	Nutzung von Datentypinformationen.....	339
7.4.1	Datentyp-Schnittstellen.....	341
7.4.2	Lokales Dictionary.....	342
7.4.3	Wertehilfen nutzen.....	342
7.4.4	Dynamische Datentypen.....	348
7.4.5	Strukturen und Stärken adaptiver RFC-Modelle.....	349

8 SAP NetWeaver Java Development Infrastructure: Komponentenmodell und Services 353

8.1	Besonderheiten großer Software-Projekte.....	353
8.1.1	Beispiel eines typischen Entwicklungsprozesses ohne zentrale Infrastruktur.....	354
8.1.2	Software-Logistik in der Java-Entwicklung.....	356
8.2	Elemente der SAP NetWeaver Java Development Infrastructure.....	358
8.2.1	Komponentenmodell der SAP.....	359
8.2.2	Design Time Repository.....	374
8.2.3	Component Build Service (C8S).....	388
8.2.4	Change Management Service (CMS).....	395
8.2.5	Software-Logistik im Entwicklungsprozess mit der SAP NetWeaver JDI.....	403

9 SAP NetWeaver Java Development Infrastructure: Step by Step zur Beispielanwendung 409

9.1	Die Employee-Beispielanwendung.....	410
9.2	Erste Schritte mit der SAP NetWeaver JDI.....	411
9.2.1	Administration der SAP NetWeaver JDI.....	412
9.2.2	Development-Configuration importieren.....	414
9.3	Datenmodell definieren.....	415
9.3.1	Dictionary-DC-Projekt anlegen.....	416
9.3.2	Employee-Tabelle definieren.....	420
9.3.3	Public Parts der Development-Components definieren.....	420
9.3.4	Development-Components bauen und deployen.....	423
9.4	Zugriff auf Tabellendaten und Business-Logik realisieren.....	424
9.4.1	EJB-Modul-DC anlegen.....	424
9.5	JSP-basiertes User Interface erstellen.....	425
9.6	J2EE-Gesamtanwendung anlegen und testen.....	428
9.7	Entwicklungsobjekte zentral zur Verfügung stellen.....	429
9.7.1	Quelldateien einchecken und alle DCs aktivieren.....	429
9.7.2	Änderungen freigeben.....	431

10 SAP NetWeaver Java Development Infrastructure: Konfiguration und Administration 433

10.1	Konfiguration der SAP NetWeaver Java Development Infrastructure	433
10.1.1	Java-Entwicklungslandschaft.....	434
10.1.2	Einrichten einer SAP NetWeaver Java Development Infrastructure.....	438
10.2	Administration der SAP NetWeaver Java Development Infrastructure.....	461
10.2.1	Produktdefinition im SLD.....	462
10.2.2	Namensraumpräfix.....	463
10.2.3	Vorbereiten eines Tracks.....	467
10.2.4	Entwicklung in der Java Development Infrastructure.....	474
10.2.5	Konsolidierungsphase.....	480
10.2.6	Montage der Software und Qualitätssicherung.....	481
10.2.7	Auslieferung an die Kunden.....	483

11 SAP Web Application Server: Architektur 485

11.1	Systemlandschaft des SAP Web Application Servers.....	486
11.1.1	Design-Konzepte des Clusters.....	487
11.1.2	Komponenten des Clusters.....	489
11.1.3	Optionen der Cluster-Systemlandschaft.....	494
11.2	Cluster Lifecycle Management.....	496
11.3	Skalierbarkeit und hohe Verfügbarkeit.....	504
11.3.1	Anfragenbearbeitung und Lastausgleich.....	504
11.3.2	Hohe Verfügbarkeit.....	508
11.3.3	Session-Failover.....	512
11.4	Remote Debugging.....	516
11.5	Laufzeitarchitektur.....	518
11.5.1	Java-Kernel.....	520
11.5.2	Infrastrukturschicht.....	523
11.5.3	Classloading und Isolierung.....	525
11.5.4	Deployment-Service und J2EE-Container.....	533

12 SAP Web Application Server: Supportability 535

12.1	Protokollierung.....	535
12.2	Überwachung/Monitoring.....	538
12.2.1	JMX-Infrastruktur.....	538
12.2.2	Monitore.....	540
12.2.3	Neue Inhalte im Monitoring Framework hinzufügen.....	543
12.2.4	Auf Monitoring-Daten von externem Client zugreifen.....	544
12.3	Administration.....	545
12.3.1	Administrationsinfrastruktur.....	545
12.3.2	Erweiterungen des Visual Administrators.....	549
12.3.3	Benutzerschnittstelle für neu installierten Service.....	549
12.3.4	Neues verwaltetes Objekt im Administrationsverzeichnisbaum .	550
12.3.5	Weitere administrative Tools.....	551
12.4	Performance-Analyse.....	552
12.4.1	Application-Tracing.....	552
12.4.2	Single Activity Trace.....	553
12.4.3	SQL-Tracing.....	554

Die Autoren 557

Index 561