

Inhaltsverzeichnis

1	Einführung	1
1.1	Warum Webanwendungen mit SAP?	1
1.2	Warum Business Server Pages?	4
1.3	Web Dynpro vs. offenes UI	5
1.4	Über dieses Buch	6
1.5	Danksagungen	10
2	Der Retail Store der Migros	11
2.1	Was ist der Retail Store?	11
2.2	Das Projekt »SRS auf BSP«	13
2.3	Das Framework	15
2.4	Zusammenfassung	17
3	Grundsätze der ABAP-Entwicklung	21
3.1	Wozu dieses Kapitel?	21
3.2	Wiederverwendbarkeit	21
3.2.1	Wiederverwendbare Module schreiben	23
3.2.2	ReUse Libraries	24
3.2.3	Refactoring	25
3.2.3.1	Faktorisieren	26
3.2.3.2	Umzug von Code in Klassen	26
3.2.3.3	Umhängen von Funktionsbausteinen	27
3.3	Modularisierung	28
3.3.1	Includes	29
3.3.2	Macros	30
3.3.3	Unterprogramme	33
3.3.3.1	Nachteile von Unterprogrammen	36

3.3.4	Funktionsbausteine	38
3.3.4.1	Schnittstelle	39
3.3.4.2	Ausnahmen	40
3.3.4.3	Testmodus	42
3.4	Entkopplung	42
3.4.1	Entkopplung mit Interfaces	43
3.4.2	Entkopplung mit Ereignissen	45
3.4.3	Ein Beispiel: Anwendungs-Traces	45
3.4.4	Schichten (Trennung UI/PD)	50
3.5	Objektorientierte Programmierung	51
3.5.1	Entwurfsprinzipien der OOP	51
3.5.1.1	Das Open Closed Principle (OCP)	51
3.5.1.2	Das Liskow Substitution Principle (LSP)	52
3.5.1.3	Das Dependency Inversion Principle (DIP)	53
3.5.1.4	Das Interface Segregation Principle (ISP)	53
3.5.2	Einige Entwurfsmuster	54
3.5.2.1	Das Singleton	54
3.5.2.2	Der abstrakte Server	55
3.5.2.3	Der Adapter	56
3.5.2.4	Das Proxy	56
3.5.2.5	Der Erbauer	57
3.5.2.6	Der Beobachter	58
3.5.3	Klassen	59
3.5.3.1	Mit dem Class Builder erstellte Klassen	60
3.5.3.2	Implementierte Interfaces	61
3.5.3.3	Lokale Klassen	62
3.5.3.4	Friends	63
3.5.4	Instanzlose Methoden (Klassenmethoden)	63
3.5.5	Objekte und ihre Erzeugung	64
3.5.5.1	Singletons	65
3.5.5.2	Klassen mit Instanzverwaltung	66
3.5.5.3	Objekterzeugung in Fabriken	68
3.5.6	Ereignisse	70
3.5.6.1	Ereignisparameter	70
3.5.6.2	Changing-Parameter von Ereignissen	71
3.5.6.3	Aktivierung der Ereignisbehandlung	72
3.5.6.4	Ereignis-Interfaces	73
3.5.7	Interfaces	73

4	Anwendungen mit Benutzerschnittstelle	75
4.1	Am Anfang war EVA ...	75
4.2	... dann kam MVC	78
4.3	Der View	81
4.4	Der Controller	82
4.5	Das Model	82
4.6	Anwendung von MVC auf Reports	83
5	HTTP-Verbindungen mit dem SAP-System	89
5.1	Ein einfacher Requesthandler	89
5.2	Der ICF-Servicebaum	94
5.3	Anmeldung	97
5.4	Das MIME Repository	101
6	MVC-basierte Webanwendungen	103
6.1	Das Framework	103
6.2	Eine einfache BSP-Applikation	105
6.3	Das Panel	109
6.4	Visuelle Elemente mit Tag Libraries erzeugen	112
6.5	Komponenten vom Viewtyp	113
6.6	Serverseitige Framesets	116
6.7	Komponenten vom Controllertyp	120
6.8	Funktionscodes	121
6.9	Ablaufschema des Controllers	125
6.10	Komponente und Panel	127
6.11	Viewkontexte	131
7	Datenbindung	135
7.1	Syntax und Funktionsweise	135
7.2	Datenbindung an Controller	138
7.3	Datenbindung an andere Objekte	138
7.4	Get-Methoden in der Datenbindung	139
7.5	Datenvalidierung	141
7.6	Datenübernahme	145
7.7	Modelübergreifende Abhängigkeiten	146

7.8	Abschlussarbeiten der Models	148
7.9	Das Interface ZIF_MODEL	149
7.10	Sichtbarkeit von Attributen	151
7.11	Änderbarkeit von Attributen (Feldauswahl)	153
7.11.1	Das Controller-Interface ZIF_INPUT_FIELD	154
7.11.2	Das Model-Interface ZIF_MODEL_CHECK	156
8	Ein praktisches Beispiel: IT-Meldungen erfassen	159
8.1	Die Geschäftslogik	159
8.1.1	Das Datenmodell	161
8.1.2	Die Funktionen – Sichern	163
8.2	Die Webanwendung	165
8.3	Validierungen im Entwicklungsmodus	166
8.4	Das Erfassungsbild	167
8.5	Fcode-Anbindung – Sichern	169
8.6	Eingebaute Validierungen	171
8.7	Validierungen der BAPIs und APIs	173
8.8	Clientseitige Validierungen – Beispiel Mussfelder	176
8.9	Serverseitige Prüfungen – validate_field()	177
8.10	Ergänzungen der API-Prüfungen	179
8.11	Meldungen anlegen und ändern	183
8.12	Liste von Meldungen	187
8.13	Anpassung der Tabellensicht	188
8.14	Die Toolbar	191
8.15	Verwendung von Sitzungsdaten	194
9	Sitzungsdaten	197
9.1	Stateful oder Stateless	197
9.2	Versteckte Formularfelder	198
9.3	Clientseitige Cookies	199
9.4	Der Puffer des Anwendungsservers	202
9.5	Serverseitige Cookies	203
9.6	Die Sitzungs-ID	204
9.7	Löschen veralteter Cookies	205
9.8	Eine Tabelle von Sitzungsdaten	205

9.9	Serialisierung	209
9.10	Der ICM-Server-Cache	211
9.11	Der Client-Cache	213
10	Der Message Handler	215
10.1	Notwendigkeit des Nachrichtensammlers	215
10.2	Die Klasse <code>zcl_messages</code>	217
10.3	Instanzbeschaffung	218
10.4	Nachrichten sammeln	219
10.5	Nachrichten lesen	221
10.6	Meldungsfiler, Positionsnummer und Aufsetzpunkt	221
10.7	Meldungen anzeigen und sichern	222
10.8	Ein Beispiel	222
11	Die Tag Library	227
11.1	Warum Tag Libraries?	227
11.2	Eigene Tag Libraries?	228
11.3	Das Interface <code>IF_BSP_ELEMENT</code>	233
11.4	<code><z:input></code> -Eingabefelder	235
11.4.1	Der Doku-Prozessor	238
11.4.2	Suchhilfeanbindung	239
11.4.3	Der Suchhilfeprozessor	240
11.4.4	Datenbindung	243
11.4.5	Feldmodifikation	244
11.5	<code><z:message></code> – eine Meldung im View ausgeben	245
11.6	<code><z:messages></code> – Mehrere Meldungen darstellen	247
11.7	<code><z:table></code> – Interne Tabellen präsentieren	248
11.7.1	Einfachstaufruf	249
11.7.2	Spalten ausblenden	250
11.7.3	Substitutionen	251
11.7.4	Exits	253
11.7.5	Blättern	255
11.7.6	Markierspalten	256
11.7.7	Die Toolbar	257
11.7.8	Weitere Attribute des <code><z:table></code> -Elements	258

11.8	Das <z:button>-Tag – Buttons im SRS-Layout	259
11.8.1	Darstellungsformen	259
11.8.2	Radiobuttonleisten	261
11.8.3	Funktionscode und Hyperlink	262
11.8.4	Der Busy-Check	263
11.8.5	Feldauswahl	264
11.9	<z:longtext> – SAPscript-Texte einbinden	264
11.10	Das <z:form>-Element	265
12	Das MVC-Framework im Detail	267
12.1	Überblick	267
12.2	Der Controller	270
12.3	Die MVC-Fabrik	273
12.4	Die Schnittstelle ZIF_MVC_FRAMEWORK	275
12.5	Model und Modelfabrik	278
12.6	Validierung und Datenbindung	279
12.7	Typische Redefinitionspunkte von ZCL_CONTROLLER	280
13	Gerätespezifisch rendern	283
13.1	Bildschirmauflösung	284
13.2	View-Ersetzungen	285
13.3	Client-Informationen	287
13.4	Browserweichen	287
14	Mehrsprachige Texte	289
14.1	Das Online Text Repository	289
14.2	Die Klasse ZCL_BSP_UTIL	293
14.3	Suche nach Texten	295
15	Clientseitige Logik	297
15.1	Formulare	297
15.1.1	Absenden von Formularen	297
15.1.2	Die Funktion genericSubmit()	299
15.1.3	Navigation mit Formularen	300
15.1.4	Dynamisch erzeugte Formulare	301
15.2	Hilfefenster	302
15.3	Eingabeprüfungen	303

15.4	Cursorsteuerung	305
15.5	Clientseitiges Rendern	306
16	Andere Requesthandler	315
16.1	XML-basierte HTTP-Requests	315
16.2	Komplexe Requests vom Browser absetzen	317
16.2.1	Microsofts XMLHTTP Objekt	318
16.2.2	Der HTTP-Requestor – ein hilfreiches Applet	320
16.3	Ajax: Das neue Web	324
16.4	Ein XML-basierter Data Browser	326
16.4.1	Datenformat für die Selektion	326
16.4.2	Programmdesign	327
16.4.3	Transformation der Eingangsdaten	328
16.4.4	Dynamische Selektion	332
16.4.5	Transformation der Rückgabeliste	333
16.4.6	Aufruf aus dem Netz	336
16.4.7	Zusammenfassung	336
16.5	Das SAP-System als HTTP-Client	338
16.5.1	HTTP-Destinationen	338
16.5.2	HTTP-Verbindungen testen	339
16.5.3	Ein RSS-Newsreader	341
16.6	Web-Services	349
16.6.1	Ein Beispiel	351
16.6.2	Wann kommen Applets in Web-Services?	356
16.7	Der Frontcontroller	357
17	Ajax im Einsatz: Eine Jobverwaltung	363
17.1	Aufgabenstellung	363
17.2	Abhängige Listboxen versorgen	364
17.3	Ermittlung von Reportvarianten	366
17.4	Den Request absetzen	370
17.5	Ausgabe von Meldungen	373
17.6	Verwaltung der Jobtabelle	374
17.7	Darstellung der Jobtabelle	378
17.8	Die HTML-Seite	379

17.9	Logik der Seite	382
17.9.1	Logik beim Laden der Seite	383
17.9.2	Logik beim Ändern von Feldinhalten	385
17.9.3	Logik der Tabelle	386
17.9.4	Logik der Drucktasten	386
17.10	Zusammenfassung	389
18	Pflege von Datenbanktabellen	391
18.1	Problem	391
18.2	Bestehende Lösungen	392
18.3	Der generische Tabellenhandler – Anforderungen	393
18.4	Datenmodell	394
18.4.1	Ändern	395
18.4.2	Löschen	395
18.4.3	Einfügen	396
18.4.4	Datenbankstand ermitteln	396
18.4.5	Ermitteln, ob Änderungen zu sichern sind	397
18.4.6	Daten sichern	397
18.5	Implementierung	397
18.5.1	Attribute von generischem Typ	397
18.5.2	Statische Struktur der Klassen	398
18.5.3	Dynamische Klassengenerierung	400
18.5.4	Der »Scope«	405
18.5.5	Verbuchung	407
18.5.6	Ereignisse	408
18.5.7	Einsatz in Anwendungen	410
19	Applikationsparameter	411
19.1	Datenmodell	412
19.1.1	Definitionsbereich	412
19.1.2	Wertebereich	413
19.2	Anforderungen an Pflege und Abfrage	414
19.3	Programmdesign	414
19.3.1	Klasse <code>zcl_param</code>	414
19.3.2	Parameterpflege im GUI	415
19.4	Webanwendung zur Parameterpflege	416

A	Anhang	419
A.1	DTD der Flow Logic (config.xml)	419
A.2	Validierung des erzeugten HTML-Codes	420
A.3	Eigene HTML-Vorlagen für Views definieren	428
A.4	Verwendungsnachweis von Texten in BSP-Views	429
A.5	Namenskonventionen für ABAP-Variablen	430
A.6	Den ICM ad hoc einrichten	431
	Glossar	433
	Literatur und Weblinks	439
	Index	441