



Einstieg in ASP.NET
Matthias Lohrer
Galileo Computing
546 S., geb., mit CD
29,90 Euro, ISBN 3-89842-302-6

Inhalt

Vorwort 17

-
- 1 Wie .NET und ASP.NET funktionieren 23**
 - 1.1 **Komponenten der .NET-Architektur 23**
 - 1.1.1 Common Language Runtime (CLR) 23
 - 1.1.2 .NET Framework 24
 - 1.1.3 Microsoft Intermediate Language (MSIL) 24
 - 1.1.4 Managed Code, unmanaged Code 25
 - 1.1.5 Common Type System (CTS) 25
 - 1.1.6 Common Language Specification (CLS) 25
 - 1.1.7 Just-in-Time-Compiler 25
 - 1.1.8 ASP.NET 25
 - 1.1.9 Web Services 26
 - 1.1.10 VB- und C#-Compiler 26
 - 1.1.11 Assembly 26
 - 1.2 **Aufbau des .NET Frameworks 26**
 - 1.2.1 Namensräume (Namespaces) 26
 - 1.2.2 Häufig verwendete Namensräume 29
 - 1.3 **Grundlagen von ASP.NET 30**
 - 1.3.1 Gemeinsamkeiten mit anderen serverseitigen Programmiersprachen 30
 - 1.3.2 Unterschiede zu anderen serverseitigen Programmiersprachen 31
-
- 2 Installation 33**
 - 2.1 **Systemvoraussetzungen 33**
 - 2.2 **Die Internet-Informationdienste installieren 34**
 - 2.3 **MDAC 2.7 installieren 35**
 - 2.4 **Das .NET Framework SDK installieren 35**
 - 2.5 **Das aktuelle Service Pack installieren 36**
 - 2.6 **Die .NET-Dokumentation aufrufen 36**
 - 2.7 **Die erste ASP.NET-Seite aufrufen 37**
-
- 3 Hinweise zu VB.NET 39**
 - 3.1 **Eine Testumgebung 39**
 - 3.2 **Variablen 40**
 - 3.2.1 Bezeichner 41
 - 3.2.2 VB.NET-Datentypen 41

3.3	Konstanten	48
3.4	Typkonvertierungen	48
3.5	Kontrollstrukturen	48
3.5.1	Entscheidungen mit If	48
3.5.2	Select Case	49
3.5.3	Zählschleife mit For Next	49
3.5.4	Schleife mit Do Loop	50
3.5.5	Schleife mit While	50
3.6	Funktionen und Prozeduren	51
3.6.1	Aufbau	51
3.6.2	Übergabe von Parametern	51
3.7	Klassen erstellen	53
3.7.1	Eigenschaften einer Klasse über Felder realisieren	54
3.7.2	Eigenschaften mit Property-Prozeduren realisieren	55
3.7.3	Schreibgeschützte Eigenschaften	56
3.7.4	Methoden erstellen	56
3.7.5	Konstruktor erstellen	56
3.7.6	Klassen vererben	57
3.8	Mit Objekten arbeiten	58
3.8.1	Objekte vergleichen	58
3.8.2	Den Typ eines Objekts bestimmen	59
3.8.3	Objekte in andere Klassen konvertieren	59
3.9	Arbeiten mit Arrays	61
3.10	Dateizugriffe	63
3.11	Fehlerbehandlung mit Try ... Catch ... Finally ... End Try	64
<hr/>		
4	»Hallo, Welt!« – Ein ASP.NET-Crashkurs	67
4.1	Der Aufbau von ASP.NET-Seiten	67
4.1.1	Die @Page-Direktive	68
4.1.2	Der serverseitige Skript-Block	69
4.1.3	Der Block mit HTML-Code	69
4.2	Das erste ASP.NET-Skript: »Hallo, Welt!«	70
4.2.1	»Hallo Welt!« – fast wie ASP	70
4.2.2	»Hallo, Welt!« im ASP.NET-Stil	71
4.2.3	»Hallo, Welt!« mit serverseitigen Steuerelementen	74
4.3	Formulare mit ASP.NET auswerten: »Hallo, Anwender!«	76
4.3.1	ASP.NET begrüßt den Anwender	76
4.3.2	Ein Grundgerüst für ASP.NET-Formulare	80
4.3.3	Validierung von Anwendereingaben	81
4.4	Zusammenfassung und Ausblick	85

5	Die Page-Klasse	87
5.1	Der Lebenszyklus einer aspx-Seite	90
5.1.1	Lebenszyklus beim ersten Aufruf	90
5.1.2	Lebenszyklus bei einem Postback	91
5.1.3	In Page_Load die Validierung ausführen	92
5.2	Die Code-Behind-Technologie	94
5.2.1	Die aspx-Seite mit der Code-Behind-Datei verknüpfen	97
5.2.2	Die Code-Behind-Datei erstellen	97
5.3	Die Page-Direktive	99
5.4	Die Controls-Auflistung	101
6	HTML-Serversteuerelemente	103
6.1	System.Web.UI.LiteralControl	105
6.2	Die HTML-Serversteuerelemente. Vererbungshierarchie und gemeinsame Eigenschaften	107
6.3	Die Klasse HtmlImage	109
6.4	HtmlContainerControl und abgeleitete Klassen	113
6.4.1	Die Basisklasse HtmlContainerControl	113
6.4.2	HtmlGenericControl	118
6.4.3	HtmlAnchor	118
6.4.4	HtmlTable, HtmlTableRow und HtmlTableCell	122
6.4.5	HtmlForm	127
6.4.6	HtmlButton	130
6.4.7	HtmlSelect	133
6.4.8	HtmlTextArea	144
6.5	HtmlInputControl und abgeleitete Klassen	146
6.5.1	Ein Beispiel mit mehreren Eingabeelementen	147
6.5.2	HtmlInputButton	150
6.5.3	HtmlInputImage	152
6.5.4	HtmlInputFile	153
6.5.5	HtmlInputText	159
6.5.6	HtmlInputCheckBox	161
6.5.7	HtmlInputRadioButton	162
6.5.8	HtmlInputHidden	165
7	Validierung von Anwendereingaben	167
7.1	RequiredFieldValidator	168
7.1.1	Display-Attribut	170
7.1.2	InitialValue-Attribut	171
7.1.3	Die Fehlermeldung definieren	175
7.2	Das ValidationSummary-Steuerelement	176

- 7.3 **Wie das Display-Attribut funktioniert** 180
- 7.4 **CompareValidator** 183
 - 7.4.1 Die Eingabe mit einem fixen Wert vergleichen 184
 - 7.4.2 Zwei Eingabefelder miteinander vergleichen 185
 - 7.4.3 Eine Typüberprüfung vornehmen 187
- 7.5 **RangeValidator** 188
- 7.6 **RegularExpressionValidator** 192
 - 7.6.1 Reguläre Ausdrücke erstellen 193
 - 7.6.2 Eingaben mit regulären Ausdrücken überprüfen 195
 - 7.6.3 Konventionen für gültige Passwörter überprüfen 197
- 7.7 **CustomValidator** 199
 - 7.7.1 Serverseitige Überprüfung 199
 - 7.7.2 Eine clientseitige Überprüfung ergänzen 202

8 **Webserversteuerelemente** 205

- 8.1 **Webserversteuerelemente formatieren** 206
 - 8.1.1 Die FontInfo-Klasse 208
 - 8.1.2 Die System.Drawing.Color-Struktur 209
 - 8.1.3 Die Klasse Unit und die Enumeration UnitType 210
- 8.2 **Button** 211
 - 8.2.1 OnClick 211
 - 8.2.2 OnCommand 213
- 8.3 **LinkButton** 214
- 8.4 **ImageButton** 218
- 8.5 **HyperLink** 220
- 8.6 **Label** 221
- 8.7 **Literal** 222
- 8.8 **Panel** 222
- 8.9 **TextBox** 223
 - 8.9.1 Mit JavaScript den Fokus setzen 227
- 8.10 **CheckBox** 230
- 8.11 **Image** 233
- 8.12 **Webserversteuerelemente für Tabellen** 234
- 8.13 **RadioButton** 239
- 8.14 **Calendar** 242
- 8.15 **AdRotator** 247
- 8.16 **E-Mails versenden** 249
 - 8.16.1 Versand von E-Mails mit der Klasse SmtpMail 249
 - 8.16.2 E-Mails mit der Klasse MailMessage erstellen 250

9 Webserversteuerelemente für die Darstellung von Listen 253

- 9.1 Die abstrakte Basisklasse ListControl 254**
- 9.2 Das ListItem-Objekt 255**
- 9.3 CheckBoxList 255**
 - 9.3.1 Kontrollkästchen-Listen statisch erstellen 256
 - 9.3.2 Kontrollkästchen-Listen dynamisch erstellen 258
 - 9.3.3 Dynamische Datenbindung 259
- 9.4 RadioButtonList 260**
- 9.5 DropDownList 263**
- 9.6 ListBox 266**
- 9.7 Repeater und DataList 270**

10 Webserversteuerelemente selbst erstellen 271

- 10.1 Benutzersteuerelemente (User Controls) 271**
 - 10.1.1 Ein einfaches Benutzersteuerelement erstellen 271
 - 10.1.2 Die Control-Direktive 273
 - 10.1.3 Die Register-Direktive für Benutzersteuerelemente verwenden 274
 - 10.1.4 Die Basisklasse UserControl 274
 - 10.1.5 Eigenschaften für Benutzersteuerelemente definieren 275
 - 10.1.6 In Benutzersteuerelementen auf Ereignisse reagieren 277
- 10.2 Benutzerdefinierte Steuerelemente (Custom Controls) 280**
 - 10.2.1 Eine zählende Schaltfläche 281

11 Der Status von Seiten, Sitzungen und Applikationen 285

- 11.1 Versteckte Felder in Formularen nutzen 285**
- 11.2 Der Status einer Seite: Viewstate 286**
 - 11.2.1 Eigene Daten im Viewstate speichern 287
 - 11.2.2 Den Anzeigestatus ein- und ausschalten 290
- 11.3 Der Status einer Sitzung: Session 290**
 - 11.3.1 Die Klasse HttpSessionState 290
 - 11.3.2 Wie ASP.NET die Probleme des klassischen ASP löst 294
 - 11.3.3 Übertragung der Session-ID: Cookie oder URL 296
 - 11.3.4 Den Sitzungsstatus ein- und ausschalten 298
 - 11.3.5 Den Sitzungsstatus konfigurieren 301
 - 11.3.6 Session-Ereignisse verwenden 302
 - 11.3.7 Objekte mit Session-Scope einbinden 305
 - 11.3.8 Instanzen eigener Klassen in den Sitzungsdaten speichern 307
- 11.4 Cookies verwenden 315**
- 11.5 Der Status einer Anwendung: Application 318**
 - 11.5.1 Den Zugriff auf anwendungsweite Daten synchronisieren 319
 - 11.5.2 Beschränkungen von Application-Variablen 320

12	ASP.NET und Datenbanken	321
12.1	Auf Datenbanken zugreifen – ein Crashkurs	321
12.1.1	Eine Tabelle im Browser anzeigen	321
12.1.2	Über den Browser einen Datensatz hinzufügen	324
12.2	Wie ADO.NET funktioniert – ein Überblick	327
12.3	Die Verbindung zur Datenbank herstellen	328
12.3.1	OleDbConnection	328
12.3.2	SqlConnection	332
12.4	Daten lesen mit SqlCommand und OleDbCommand	336
12.4.1	ExecuteReader	336
12.4.2	ExecuteNonQuery und die Parameters-Collection	337
12.4.3	ExecuteScalar	338
12.5	SqlDataReader und OleDbDataReader	341
12.5.1	Ein Reader-Objekt an ein Steuerelement binden	342
12.5.2	Ein Reader-Objekt an ein Listensteuerelement binden	342
12.5.3	Ein Reader-Objekt zeilenweise auswerten	345
12.6	Die DataSet-Klasse	353
12.6.1	Das Zusammenspiel von Command, Adapter und DataSet	354
12.6.2	Mehrere Tabellen in ein DataSet-Objekt einlesen	357
12.6.3	Auf einzelne Zeilen, Spalten und Tabellen eines DataSet-Objekts gezielt zugreifen	361
12.6.4	Relationen zwischen Tabellen festlegen	368
12.7	DataViews verwenden	372
12.7.1	Tabellen sortieren	373
12.7.2	Tabellen nach Inhalten filtern	374
12.7.3	Nach dem Zeilenstatus filtern	374
12.7.4	In Tabellen suchen	375
12.8	Mit ASP.NET Daten bearbeiten	378
12.9	Daten bearbeiten mit dem Command-Objekt	379
12.10	In-Place-Editing mit dem DataGrid-Steuerelement	382
12.10.1	Die EditItemIndex-Eigenschaft	382
12.10.2	Das asp:DataGrid-Tag anpassen	384
12.10.3	OnEditCommand: Daten bearbeiten	387
12.10.4	OnCancelCommand: Die Bearbeitung abbrechen	391
12.10.5	OnUpdateCommand: Die Änderungen sichern	391
12.10.6	Das Repeater-Steuerelement verwenden	397
12.10.7	Die Ereignisse des Repeater-Steuerelements auswerten	400
12.11	Das DataList-Steuerelement	403
12.11.1	In-Place-Editing mit dem DataList-Steuerelement	404

-
- 13 »Wer darf was?« ASP.NET und Sicherheit 413**
 - 13.1 **»Wer darf was?« mit Windows-Bordmitteln entscheiden 414**
 - 13.1.1 Windows, Standardauthentifizierung 414
 - 13.1.2 Windows, Digestauthentifizierung 414
 - 13.1.3 Integrierte Windows-Authentifizierung 415
 - 13.1.4 Beispiel für die Verwendung der Windows-Standardauthentifizierung 415
 - 13.1.5 Dateibasierte Autorisierung über ACL verwenden 419
 - 13.1.6 Kriterien für die Auswahl einer Authentifizierungsmethode 422
 - 13.2 **»Wer darf was?« innerhalb von ASP.NET entscheiden 423**
 - 13.2.1 Die IIS für formularbasierte Authentifizierung einrichten 424
 - 13.2.2 Eine zu schützende Datei erstellen 425
 - 13.2.3 In der Datei web.config die formularbasierte Authentifizierung einrichten 425
 - 13.2.4 In einer weiteren web.config-Datei die Autorisierung einrichten 426
 - 13.2.5 Die Datei mit dem Login-Formular erstellen 426
 - 13.2.6 Die Zugangsdaten in der web.config speichern 429
 - 13.2.7 Den authentifizierten Anwender erkennen und ein Logout ermöglichen 431
 - 13.2.8 Passwörter in der web.config in verschlüsselter Form speichern 434

14 Webanwendungen erstellen und konfigurieren 437

- 14.1 **Eine Webanwendung neu erstellen 437**
 - 14.1.1 Ein physikalisches Verzeichnis festlegen 437
 - 14.1.2 Eine Start-Datei erstellen 437
 - 14.1.3 Ein virtuelles Verzeichnis definieren 438
 - 14.1.4 Die Startseite festlegen 441
- 14.2 **Die Webanwendung mit global.asax konfigurieren 443**
 - 14.2.1 Direktiven 444
 - 14.2.2 Der Skriptblock 445
 - 14.2.3 Server Side Includes 449
 - 14.2.4 Objektdeklarationen 450
 - 14.2.5 Die Klasse HttpApplication 451
 - 14.2.6 Eine Webanwendung starten/beenden/neu starten 452
- 14.3 **Die Konfigurationskaskade mit machine.config und web.config 452**
 - 14.3.1 Konfiguration von virtuellen und von physikalischen Verzeichnissen 453
 - 14.3.2 Das Format von web.config 453
 - 14.3.3 Der Aufbau von web.config 454
 - 14.3.4 Konfigurationsabschnitte 457
 - 14.3.5 Eigene Konfigurationen erstellen 461
 - 14.3.6 Vorteile der hierarchischen Konfiguration 465

15	Leistungssteigerung durch Caching	467
15.1	Ganze Seiten cachen	468
15.1.1	Formulare cachen	470
15.2	Teile von Seiten cachen	472
15.2.1	Benutzersteuerelemente cachen, die Formulare enthalten	474
15.3	Cache-Einstellungen mit der Klasse <code>HttpCachePolicy</code> programmgesteuert beeinflussen	476
15.4	Beliebige Daten und Objekte im Cache ablegen	478
15.5	Den Cache automatisch aktualisieren	482
15.5.1	Den Cache vom Inhalt einer XML-Datei abhängig machen	482
15.5.2	Weitere Möglichkeiten, den Cache abhängig zu machen	485
15.5.3	Den Cache über Prioritäten steuern	486
15.5.4	Reagieren, wenn ein Eintrag aus dem Cache gelöscht wird	487
15.6	Allgemeine Tipps zur Steigerung der Performance	491
15.7	Weiterführende Ressourcen	492
16	Web Services	493
16.1	Wie Web Services funktionieren	493
16.1.1	Basistechnologie HTTP	494
16.1.2	Basistechnologie XML	494
16.1.3	Basistechnologie SOAP	495
16.1.4	Umdenken beim Entwickeln von Web Services	495
16.2	Einen einfachen Web Service erstellen und anbieten	496
16.2.1	Die Datei <code>webservice01.asmx</code> erstellen	496
16.2.2	<code>webservice01.asmx</code> im Browser aufrufen	498
16.3	Einen Web Service verwenden	501
16.3.1	Eine Proxyklasse erstellen	502
16.4	Einen Web Service mit einem SOAP-Header sichern	507
16.4.1	Einen passwortgeschützten Web Service erstellen	508
16.4.2	Die Proxyklasse erzeugen	509
16.4.3	Den passwortgeschützten Web Service aufrufen	510
16.4.4	Fazit: Web Services und Sicherheit	512
16.5	Ein <code>DataSet</code>-Objekt übertragen	512
16.5.1	Den Web Service in einer <code>asmx</code> -Datei erstellen	512
16.5.2	Die Proxyklasse erzeugen	514
16.6	Web Services finden	516
16.6.1	Discovery	516
16.6.2	UDDI	520
16.7	Zusammenfassung und Ausblick	521
16.7.1	Zusammenfassung	521

- 16.7.2 Ausblick 521
- 16.7.3 Ressourcen 521

A Anhang 523

- A.1 Provider mit ASP.NET-Unterstützung 523
- A.2 Websites mit Know-how zu ASP.NET 523
- A.3 Die CD-ROM zum Buch 524

Index 525