

Web 2.0 mit ASP.NET 4.0 und AJAX

Praxiseinstieg in die Erstellung
interaktiver Websites



Einführung

Um das Einfachste, was über »Web 2.0« gesagt werden kann, vorwegzunehmen: Der Begriff bezeichnet kein neues Internet, was sich alleine aus der Tatsache ergibt, dass das Internet aus mehr als »nur« zwischenzeitlich circa 80 Millionen Webseiten besteht. Nicht einmal eine neue Generation des World Wide Web wird unter dem Schlagwort »Web 2.0« zusammengefasst. Was aber dann? Und woher stammt die zwischenzeitlich in aller Munde befindliche Begrifflichkeit? Zunächst zur zweiten Frage: Erstmals fiel der Begriff bei einem Gedankenaustausch zwischen Tim O'Reilly und der Firma Media Live International, bei dem es um die erfolgreichen Strategien der neuen Internetfirmen ging, jene also, die nach dem Platzen der vormals so dicken Internetblase gegründet wurden. Einige Jahre später erinnert sich O'Reilly in einem Artikel (deutsche Übersetzung unter <http://www.twozero.uni-koeln.de/content/e14/index.ger.html>) mit folgenden Worten an jenes folgenreiche Meeting:

»Interessante neue Anwendungsmöglichkeiten tauchten mit erstaunlicher Regelmäßigkeit auf und die überlebenden Firmen schienen einige wichtige Dinge gemeinsam zu haben. Könnte es sein, dass der Dot-Com-Kollaps einen derartigen Wendepunkt markiert hatte, dass man diese Dinge nur mit einem Schlagwort wie Web 2.0 bezeichnen durfte? Wir einigten uns darauf und damit war die Web 2.0-Konferenz geboren.«

Web 2.0 ohne Web 1.0?

Ein paar Unbestimmtheiten finden sich in O'Reillys Zitat, und seine Ad-hoc-Begriffsschöpfung weckt eher Assoziationen mit dem Versionssprung einer kommerziellen Software denn mit dem World Wide Web. Dennoch liegt Tim O'Reilly mit der Wahl dieser Bezeichnung nicht grundsätzlich falsch, ist der zentrale Punkt beim Web 2.0 doch darin zu sehen, dem Betrachter einer Internet-Seite die Möglichkeit zu geben, aktiv Inhalte beizusteuern (auch im Sinne des Designs). Um es direkter und bewusst überspitzt zu formulieren: Der User selbst bestimmt, was er sieht, nicht (mehr) primär der Webprogrammierer/Webdesigner beziehungsweise der Betreiber einer Internetpräsenz. Es geht um nichts weniger als ein grundlegend neues Verständnis der Rolle desjenigen, der das World Wide Web nutzt.

Die Entwicklung hin zu diesem Punkt hat jedoch schon eingesetzt, lange bevor der Begriff Web 2.0 geprägt wurde. Erst durch die Einbeziehung neuerer Techno-

logien aber, wie ASP.NET auf der Serverseite, vor allem aber durch AJAX auf der Seite des Clients, wurde die Erstellung einer gänzlich neuen Klasse von Internetseiten ermöglicht. Seiten, die mit dem Submit-Reload-Muster brechen, und dem Nutzer eher den Eindruck vermitteln, mit einer typischen Desktop-Anwendung als dem World Wide Web zu arbeiten.

Auch das ist Web 2.0!

Der Aufbau des Buches

Im Wesentlichen soll anhand eines sukzessive zu entwickelnden Beispiels der Weg hin zur einer (mit Einschränkung) interaktiven Website ausführlich beschrieben werden.

Das Buch besteht aus zwei Teilen. Der kleinere erste Teil gibt Ihnen eine kurze Einführung in die grundlegenden Technologien, die für das Praxisbeispiel benötigt werden. Wenn Sie diese Grundlagen bereits beherrschen, können Sie diesen Teil auch überspringen. Der zweite Teil bildet den Hauptteil des Buches und erläutert die Beispielanwendung.

Was die in Teil 1 konzentrierte Theorie anbelangt, so habe ich mich um ein Maß bemüht, das vor allem den Neueinsteiger berücksichtigt. Während Sie in Kapitel 1 an die (technischen) Konzepte hinter AJAX herangeführt werden, gewährt Ihnen Kapitel 2 einen gründlichen Blick auf das, was sich hinter der Webbrowsererweiterung *Silverlight* verbirgt. Denn bei Silverlight geht es um bedeutend mehr als nur um ein Plug-in.

Schließlich versucht Kapitel 3, im Sinne eines Frameworks, den Schulterschluss zwischen ASP.NET und AJAX, macht es gerade im Umfeld von ASP.NET doch Sinn, den Einsatz von AJAX weitgehend zu vereinfachen. Und da existiert mehr als eine Möglichkeit ...

Weitergehende Informationen zu ASP.NET selbst erwarten Sie im einigermaßen umfangreichen Anhang A – auch solche, die teilweise weitergehen, als es die grundlegende Einführung in eine nicht ganz einfache Technologie tut.

Der zweite und wichtigste Teil des Buches: das als Buch-Community konzipierte Praxisbeispiel »Bookalike«. Die kleine Webapplikation versetzt Sie u.a. in die Lage Buchvorschläge eingeben und sich anzeigen zu lassen, Bücher zu besprechen, Profil- und Buchcover-Images auf den Server zu schicken, alles das vor dem Hintergrund einer Benutzerverwaltung. Auf der Startseite finden Sie vier ausgewählte Bücher nebst Cover, die ohne vollständiges Neuladen der Anwendung einzeln ein- und ausgeblendet werden können. Natürlich ist es auch möglich aus »Bookalike« heraus eine E-Mail zu verschicken, wahlweise im Text- oder HTML-Format. Und sollte Ihnen der Sinn danach stehen die Überschrift der einzelnen Module der

Webseite selber zu bestimmen, können Sie dies für die Bereiche »Kontakt« und »Buchvorschlag« auch tun.

Mir war es ein besonderes Anliegen, Ihnen grundlegende ASP.NET-, AJAX- und Silverlight-Funktionen im Zusammenwirken und vor dem thematischen Hintergrund der Entwicklung einer Community näherzubringen. Dabei geht es auch um das Aufzeigen konzeptioneller Schnittmengen zwischen den verwandten Technologien und Sprachen. Denn eines sollte bereits nach dem Studium der Theoriekapitel deutlich geworden sein: Eine strikte, funktionale Trennung zwischen ASP.NET, AJAX und Silverlight existiert nicht. Alleine schon deswegen, weil bei jeder der Technologien JavaScript eine entscheidende, im Falle von AJAX und Silverlight sogar *die* entscheidende Rolle spielt.

Der Umgang mit den Entwicklungsumgebungen (Visual Studio 2010 und Visual Web Developer 2010 Express) wird Ihnen in diesem Buch nicht im Sinne des beliebten Schnelleinstiegs erklärt, sondern »simultan« zur eigentlichen Entwicklungsarbeit an der Website – einer Buchcommunity – erfolgen. Learning by doing also. Mit fortschreitendem Entwicklungsstand jedoch werden die Erklärungen sowohl zur IDE als auch zu ASP.NET (sofern dies wirklich streng unterschieden werden kann) weniger werden. Nicht aus Zeit-, Platz- oder Motivationsmangel, sondern weil Sie sicher immer routinierter im Umgang mit der Entwicklungsumgebung im Allgemeinen und den Steuerelementen im Speziellen werden.

Wichtig

Wo sich Unterschiede in der programmatischen Umsetzung des Praxisbeispiels zwischen **Visual Studio 2010** und **Visual Web Developer 2010 Express** ergeben, werden Sie explizit, zu Anfang der betreffenden Kapitel, darauf aufmerksam gemacht.

Beendet sei die Vorstellung des Buchaufbaus mit dem Hinweis auf die Anhänge B und C. Auch Ihnen wird der Apache-Webserver kein Unbekannter sein. Wie Sie ASP.NET auf Grundlage des weit verbreiteten Apachen einsetzen; darüber informiert Anhang B. Last, but not least gibt Ihnen Anhang C die Möglichkeit, in C# geschriebene Codes nach VisualBasic.NET zu konvertieren, entweder »manuell« oder mit entsprechenden (Web-)Programmen – was eine gute Überleitung zum nächsten Unterkapitel ist.

Die Wahl der Programmiersprache

Hier konnte nur zwischen VisualBasic.NET und C# entschieden werden, was sich alleine aus der enormen Verbreitung beider Sprachen im Umfeld der .NET-Technologie ergibt. Dass die Wahl schließlich auf C# gefallen ist, dafür gibt es gleich mehrere Gründe:

- Bei C# existieren syntaktisch-semantische Anleihen u.a. bei C bzw. C++ als auch bei SQL, Java und Delphi. Primär jedoch geht C# auf C bzw. C++ zurück. Beides sind Sprachen mit einer beträchtlichen Verbreitung – auch was die industrielle Nutzung anbelangt.
- Ganz gleich, ob Sie dem Java-, C- oder C++-Lager entstammen. Der Umstieg auf C# wird Ihnen aufgrund des erstgenannten Punktes leichter fallen als auf VisualBasic.NET.
- C# bietet bessere Interaktionsmöglichkeiten mit nicht verwaltetem (*unmanaged*) Code, beispielsweise beim Einbinden von C-DLLs.

Eine ausführliche, strukturierte Einführung sämtlicher Sprachelemente von C# kann Ihnen dieses Buch nicht geben. Anderenfalls hätten beim Umfang des komplexen Fallbeispiels Abstriche gemacht werden müssen. Gleichwohl werden Listings in C# möglichst detailliert erläutert, streckenweise auch so weit, dass die Codierung, und damit die exemplarische Webseite, leicht eigenen Vorstellungen angepasst werden kann.

Die Zielgruppe des Buches

Computerbücher werden für Menschen geschrieben, die sich für Computer interessieren. Ob das Interesse beruflich, privat oder durch beides begründet ist, spielt dabei keine Rolle. Gleichwohl ergibt sich auch bei diesem Buch die Notwendigkeit der Definition einer wenigstens ungefähren Zielgruppe.

Bestenfalls richtet sich dieses Buch an Leser, die bereits erste Erfahrung in der Entwicklung unter .NET 3.0/3.5 (vielleicht auf sprachlicher Grundlage von VisualBasic.NET) gesammelt haben.

Aber auch dem Ein- und Umsteiger soll anhand der vorliegenden Lektüre ein gangbares Türchen geöffnet werden, woraus sich die nicht ganz einfache Notwendigkeit ergibt, die wesentlichen Konzepte von .NET, AJAX und Silverlight so zu erschließen, dass möglichst wenig Grundlegendes ausgespart, zugleich aber der Praxischarakter des Buches erhalten bleibt, und darüber hinaus ein grundsätzliches Verständnis für die vorgestellten Technologien entwickelt wird.

Wenn Sie bereits Erfahrung mit objektorientierter Programmierung, idealerweise in C, C++ oder PHP gesammelt haben, werden Sie nicht nur schneller den Einstieg in die riesigen .NET-Bibliotheken finden, sondern auch die zentralen Konzepte vor allem hinter dem mächtigen Framework (in der Version 4.0) verstehen.

HTML-Expertise in der einen oder anderen Form werden Sie mit größter Wahrscheinlichkeit besitzen. Eine gute Grundlage, um sich mit ASP.NET-Markups sowie der X(A)ML-Notation (XML für die Web-Konfigurationsdatei, XAML als Markupbasissprache für Silverlight) erfolgreich vertraut machen zu können.

Optimal wäre es, wenn Ihnen grundlegende Datenbankabfragen in SQL oder MySQL nebst entsprechender Typisierung kein unbekanntes Land wären. SQL-Abfragen unterscheiden sich von MySQL-Abfragen nur geringfügig – aber sie unterscheiden sich. Die Transformation zwischen SQL und MySQL ist gleichwohl nicht schwierig, was wiederum dazu führt, dass Erfahrungen in lediglich einer Datenbankabfragesprache ausreichend sind.

In eigener Sache

Beim Abfassen der Buchseiten habe ich mich bemüht, die deutsche Sprache nicht gänzlich zu vergessen. Im Falle der Fachbegriffe kein einfaches Unterfangen, vor allem nicht auf einem wissenschaftlichen Terrain, das geradezu übersät ist mit Anglizismen, und solchen, die zu Abkürzungen zusammengefasst werden.

Die Beherrschung der »Verenglichung« des computerbezogenen Fachterminus wird gerne als Zeichen von Professionalität und Internationalität gewertet. Letzteres mit sicher mehr Berechtigung. Schwierig wird es bei englisch lautenden Fachbegriffen, für deren Übersetzung ein deutsches Wort allein nicht ausreicht. Oder fällt Ihnen auf Antrieb ein bedeutungsgleiches Wort für »Host«, »Server« oder »Code« ein? Ja, und selbstverständlich haben englische Fachausdrücke auch die Funktion, komplexe, technische Abläufe bzw. Zusammenhänge in einem Wort zu bündeln – was vollkommen in Ordnung und sinnvoll ist.

Beispielhaft geschrieben lässt dieses Buch jedoch das Argument nicht unbedingt gelten, dass – um ein weiteres Beispiel zu bemühen – die Aura des Wortes *Workflow* eine andere ist als jene der möglichen Übersetzung »Arbeitsablauf«. So kann denn – um ein letztes Beispiel zu verwenden – statt *unmanaged* Code durchaus *unverwalteter* bzw. *nicht verwalteter*, statt *managed* Code auch *verwalteter* Code geschrieben werden – und wird es auch.

Ausdruck bester Sitte und Manier scheint seit einigen Jahren der findig sympathische Hinweis auf die Geschlechtsneutralität von Formulierungen wie »der Entwickler«, »der Programmierer« oder »der Leser« zu sein. Liebe Leserinnen: Die kurze Spanne, in der die »Computerei« eine ausschließliche Domäne des Mannes war, ist glücklicherweise schon sehr lange vorbei. Die Geschichte der Softwareentwicklung, der zahlreichen innovativen Teams (auch unter Microsoft) genauer betrachtet, gab es eine solche Phase nie. Von daher kann ich mir sicher sein, Sie werden sich bei jedem »Programmierer«, »Entwickler« und »Leser«, der Ihnen in diesem Buch als Anredeform begegnet, auf das Selbstverständlichste angesprochen fühlen.

Gerne können Sie mir Ihre Kritik und Anregungen – und natürlich auch, was Ihnen gefällt – schreiben. Die E-Mail-Adresse lautet: uklappert@web.de.