



ASP.NET voor ASP-ontwikkelaars

WAAR TE BEGINNEN?

ASP.NET scoort op vrijwel alle punten beter dan z'n voorganger ASP. ASP.NET is sneller, beter schaalbaar en stabiel. De triviale zaken die in ASP voor behoorlijke hoofdbrekens konden zorgen zijn in ASP.NET in een handomdraai gemaakt. De vraag voor de meeste ASP-programmeurs is echter: "waar moet ik beginnen?" Dit artikel probeert die vraag te beantwoorden en vertelt je tevens wat de belangrijkste redenen zijn om vandaag nog over te stappen op ASP.NET.

Het grootste verschil tussen ASP en ASP.NET – en het grootste obstakel voor ASP-programmeurs – is het programmeermodel. In ASP wordt de code in een pagina van begin tot eind uitgevoerd en moet je alle HTML vanuit die code opbouwen. Dit model is simpel, maar heeft als nadeel dat code en HTML meestal door elkaar staan. In ASP.NET is de code van de HTML gescheiden en zet je in de HTML zogenaamde server controls. Server controls zijn dynamische elementen waarvan de inhoud en eigenschappen op de server worden bepaald. Je kunt hierbij denken aan een drop-down list met gegevens uit een database, maar ook aan complexere elementen, zoals een kalender of een tabel met gegevens waar je doorheen kunt bladeren. Aan controls koppel je code die wordt uitgevoerd als de gebruiker een bepaalde actie onderneemt, bijvoorbeeld als de gebruiker in een kalender een andere dag selecteert. ASP.NET zorgt er voor dat als een dergelijke gebeurtenis optreedt, de juiste code wordt uitgevoerd. Afbeelding 1 geeft je een idee van hoe een ASP.NET -pagina met zulke functionaliteit eruit ziet.

Afbeelding 2 laat het resultaat zien nadat een gebruiker een datum heeft geselecteerd.

In afbeelding 1 is de scheiding tussen code en HTML zeer duidelijk, aangezien de code op het eerste gezicht alleen tussen server `<script>` tags staat. Dat komt omdat controls een HTML-achtige syntax hebben. Je kunt ze echter herkennen aan het attribuut `runat="server"`. Alle elementen met dit attribuut worden vanaf de server opgebouwd, waarbij gewone HTML (zonder ActiveX controls of Java-applets) gegenereerd wordt. Dit is goed te zien aan de `asp:calendar` control in afbeelding 1, die in afbeelding 2 een volledige kalender laat zien. De eigenschappen van de kalender, zoals lettertypes en kleuren, worden bepaald door attributen en eventuele subelementen van de control. In tegenstelling tot veel HTML-elementen is de benaming van attributen en subelementen van controls consistent, waardoor het onthouden ervan een stuk makkelijker is. Bovendien zijn eigenschappen van controls in een designomgeving zoals in Visual Studio.NET,

ASP.NET Web Matrix en Dreamweaver MX makkelijk in te stellen zonder dat je de eigenschappen uit je hoofd kent. Een dergelijke designomgeving en de HTML-achtige syntax maken controls ook voor webdesigners makkelijker te gebruiken. Bovendien is de taak van de webdesigner en de programmeur beter te scheiden.

In afbeelding 1 geeft het `OnSelectionChanged` attribuut van de kalender aan welke code wordt uitgevoerd als de gebruiker een andere datum selecteert. Dit lijkt erg veel op JavaScript in de browser, met het grote verschil dat de code hier wordt uitgevoerd op de server in plaats van in de browser. In de gekoppelde procedure `Calendar1_SelectionChanged` krijgt de control `label1` een waarde met daarin de geselecteerde datum. Het is belangrijk om op te merken dat deze code alleen wordt uitgevoerd als de gebruiker een andere datum selecteert. Als er andere controls zijn waarmee de gebruiker iets doet, dan wordt deze code niet uitgevoerd en blijft de waarde van de kalender en het label hetzelfde. In ASP.NET hoef je dus niet de

```

<%@ Page Language="VB" Culture="nl-NL" %>
<script runat="server">
    Sub Calendar1_SelectionChanged(sender As Object, e As EventArgs)
        Labell.Text = "Geselecteerde dag: " & _
            Calendar1.SelectedDate.ToLongDateString()
    End Sub
</script>
<html>
<head>
</head>
<body>
    <form runat="server">
        <p>
            <asp:Calendar id="Calendar1" runat="server" backcolor="#C0C0FF"
                font-size="X-Large" font-names="Verdana,Helvetica"
                OnSelectionChanged="Calendar1_SelectionChanged">
                <SelectedDayStyle backcolor="Red"></SelectedDayStyle>
                <TitleStyle backcolor="#8080FF"></TitleStyle>
                <OtherMonthDayStyle forecolor="White"></OtherMonthDayStyle>
            </asp:Calendar>
        </p>
        <p>
            <asp:Label id="Labell" runat="server"
                Font-Size="Large" Font-Names="Verdana,Helvetica">
            </asp:Label>
        </p>
    </form>
</body>
</html>

```

Afbeelding 1. Pagina met een kalender

hele HTML opnieuw op te bouwen, maar is de volgorde van gebeurtenissen des te belangrijker.

Werken met databases

Een andere grote verandering waar je onmiddellijk mee te maken krijgt is de manier waarop je met databases werkt. De objecten voor databaseoperaties zijn behoorlijk veranderd, en veel controls ondersteunen een mechanisme dat databinding heet. Met databinding koppel je een gegevensbron aan een control. De control zorgt verder dat de gegevens juist worden weergegeven, zodat je niet meer door rijen gegevens hoeft te bladeren. Afbeelding 3 laat zien hoe eenvoudig dit is met de DataGrid control, een control voor de weergave van rijen gegevens in een tabelstructuur. Met de code in afbeelding 3 worden gegevens per tien rijen afgebeeld en kan de gebruiker door de pagina's bladeren.

Het resultaat van de code in afbeelding 3 kun je zien in afbeelding 4.

De HTML in afbeelding 3 is erg kort. Dit komt omdat de opmaak van de DataGrid control hier eenvoudig is. Deze opmaak is ook niet zo van belang, omdat die makkelijk is aan te passen en uit te breiden. De code die de gegevensbron aanspreekt en via databinding koppelt aan de DataGrid is belangrijker. De databinding-code staat in een aparte procedure (`BindGrid`), zodat het vanuit verschillende gebeurtenissen aangeroepen kan worden. De eerste vijf regels van de `BindGrid` procedure zijn allemaal variabele declaraties, waarvan de eerste twee regels strings declareren met daarin de databaseconnectionstring en het SQL-statement voor de selectie van de gegevens. De derde regel declareert en instantieert een nieuw `SqlConnection`-object. Dit object is min of meer gelijk



Afbeelding 2. Resultaat van de code in afbeelding 1

aan het `ADODB.Connection`-object, maar is specifiek bedoeld om met SQL Server (of MSDE) te werken. Het `SqlConnection`-object maakt deel uit van de `System.Data.SqlClient` namespace, die daarom geïmporteerd wordt op regel 3 van afbeelding 3. Het `SqlDataAdapter`-object op de vierde regel van de `BindGrid` procedure is verantwoordelijk voor de selectie van de gegevens die moeten worden weergegeven.

Noot: Voor MS Access of andere databases, moet je de objecten in de `System.Data.OleDb` namespace gebruiken. De objecten daarin zijn nagenoeg gelijk aan die in `System.Data.SqlClient`, maar alle namen beginnen met `OleDb` in plaats van `Sql`.

DataSet, BindGrid, PageLoad

Als laatste wordt een `DataSet`-object `ds` gedeclareerd en geïnstantieerd. Het `DataSet`-object is onderdeel van de op regel 2 geïmporteerde namespace `System.Data`. Hierin bevinden zich gegevensbronafhankelijke objecten voor de manipulatie van gegevens. De `DataSet` wordt gebruikt om gegevens die zijn geselecteerd met `oCommand` op te slaan in het geheugen, waarna de database weer kan worden gesloten voordat de gegevens gebruikt worden. Het openen van de database, vullen van de `DataSet`, en het afsluiten van de database gebeurt automatisch door `oCommand.Fill(ds)`. Daarom zie je hiervoor geen afzonderlijke opdrachten. De op één na laatste regel van `BindGrid` geeft aan welke gegevensbron de DataGrid moet

```

<%@ Page Language="VB" %>
<%@ import Namespace="System.Data" %>
<%@ import Namespace="System.Data.SqlClient" %>
<script runat="server">
Sub Page_Load(Sender As Object, E As EventArgs)
    If Not Page.IsPostBack Then
        BindGrid()
    End If
End Sub

Sub DataGrid_Page(Sender As Object, _
    e As DataGridPageChangedEventArgs)
    DataGrid1.CurrentPageIndex = e.NewPageIndex
    BindGrid()
End Sub

Sub BindGrid()
    Dim dbString As String = _
        "server=(local)\NetSDK;database=Northwind;trusted_connection=true"
    Dim sqlString As String = _
        "SELECT CompanyName, ContactName FROM Customers"
    Dim oConn As New SqlConnection(dbString)
    Dim oCommand As New SqlDataAdapter(sqlString, oConn)
    Dim ds As New DataSet()

    oCommand.Fill(ds)
    DataGrid1.DataSource = ds
    DataGrid1.DataBind()
End Sub
</script>
<html>
<head>
</head>
<body>
    <form runat="server">
        <asp:datagrid id="DataGrid1" runat="server" Font-Size="X-Large"
            AllowPaging="true" OnPageIndexChanged="DataGrid_Page">
            <PagerStyle mode="NumericPages" BackColor="#C0C0FF" />
            <HeaderStyle BackColor="#C0C0FF" />
        </asp:datagrid>
    </form>
</body>
</html>

```

Afbeelding 3. Pagina met een database-selectie met paginering

gebruiken, in dit geval `ds`. De regel `DataGrid1.DataBind()` zorgt vervolgens dat de DataGrid gevuld wordt met de gegevens uit die gegevensbron.

Wanneer de pagina voor het eerst opgevraagd wordt door de gebruiker, wordt `BindGrid` aangeroepen vanuit de code die is gekoppeld aan de

`Page_Load` gebeurtenis. Deze code wordt altijd uitgevoerd, omdat deze gebeurtenis altijd plaatsvindt. Vandaar dat gekeken wordt of sprake is van een postback voordat `BindGrid` wordt aangeroepen. Er is sprake van een postback als een pagina wordt herladen doordat de gebruiker iets gedaan heeft, zoals het opvragen van een

Company Name	Contact Name
Alfreds Futterkiste	Maria Anders
Ana Trujillo Emparedados y helados	Ana Trujillo
Antonio Moreno Taqueria	Antonio Moreno
Around the Horn	Thomas Hardy
Berglunds snabbkop	Christina Berglund
Blaauw Sea Delicatessen	Hanna Moos
Bonnesdél père et fils	Frédérique Citeaux
Bólido Comidas preparadas	Marta Sommer
Bon app'	Laurence Labbon
Bottom-Dollar Markets	Elizabeth Lincoln

Afbeelding 4. Resultaat van de code in afbeelding 3

andere pagina in de DataGrid. Als dat gebeurt moeten de gegevens pas worden gekoppeld aan de DataGrid nadat de nieuwe pagina bekend is. Dit is helaas nog niet het geval als `Page_Load` wordt uitgevoerd, maar de gebeurtenis `OnPageIndexChanged` is hier speciaal voor bedoeld. De code die is gekoppeld aan `OnPageIndexChanged` stelt eerst de juiste pagina in op basis van de `DataGridPageChangedEventArgs`. Dit zijn gegevens die specifiek worden opgebouwd door de DataGrid als de gebeurtenis `OnPageIndexChanged` plaatsvindt, en bevat onder andere het nieuwe paginanummer. Als het paginanummer is aangepast, wordt `BindGrid` aangeroepen om de gegevens weer aan de DataGrid te koppelen.

Productief platform

Afbeelding 3 laat zien dat je in een handomdraai betrekkelijk complexe functionaliteit kunt maken. In ASP was het geheel een stuk ingewikkelder geweest, en was het veel lastiger om de opmaak aan te passen. De controls-architectuur van ASP.NET maakt dit alles veel eenvoudiger. Het mooie is dat je daarbij niet gelimiteerd bent tot de controls die standaard met ASP.NET worden meegeleverd. Er zijn nu al legio leveranciers die server controls aanbieden en je kunt ook zelf controls maken. In tegenstelling tot componenten die je met ASP wilt gebruiken, hoeft je de controls niet te registreren op de server. Je installeert een control gewoon door de assembly of DLL waarvan deze deel uitmaakt naar de bin-map van je webapplicatie te kopiëren, en dit kan dus zonder interventie van

hosting providers. Dit en alle andere voordelen maakt ASP.NET tot een zeer productief platform.

10 Redenen om over te stappen op ASP.NET

1. Objectgeoriënteerd programmeermodel

Zoals je hebt kunnen zien werkt ASP.NET met controls (objecten) en gebeurtenissen. Dit bevordert de productiviteit enorm en geeft bovendien betere mogelijkheden tot het hergebruiken van code.

2. Scheiding opmaak en functionaliteit

Dankzij de controls is het veel makkelijker de functionaliteit van de opmaak te scheiden. Code van een pagina kun je ook nog eens in een apart bestand zetten (dit heet code-behind), zodat designer en programmeur goed samen kunnen werken.

3. Browseronafhankelijk

De HTML die controls genereert wordt afgestemd op de browser die de pagina opvraagt. Voor een oudere browser wordt dus bijvoorbeeld pure HTML 3.2 zonder CSS genereerd.

4. Snelheid

ASP.NET is tot ruim 300% sneller dan ASP, en het laat ook andere platformen ver achter zich (zie afbeelding 5). Bovendien biedt ASP.NET allerlei mogelijkheden om pagina's of gedeeltes daarvan in het geheugen te cachen, om gebruikers nog sneller toegang te geven.

5. Stabiliteit

Bovenop de toch al stabiele omgeving van de CLR detecteert ASP.NET geheugenlekken, vastgelopen threads en andere problemen, en verhelpt ze.

6. Schaalbaarheid

Veel grotere websites worden gehost op meerdere servers (een webfarm). ASP.NET heeft onder andere het Session-object zo aangepast dat het geen probleem is om sessie-informatie te delen over meerdere servers.

7. Installatie & Updates

Het installeren of bijwerken van een applicatie is net zo eenvoudig als het kopiëren van alle bestanden. Registreren van componenten en dergelijke is niet meer nodig, en alle bestanden (ook componenten) kunnen altijd overschreven worden; ook als ze in gebruik zijn.

8. Configuratie

Alle instellingen van een applicatie staan in het bestand web.config. Dit is een XML-bestand dat je eenvoudig naar een andere server kunt kopiëren om de instellingen te gebruiken.

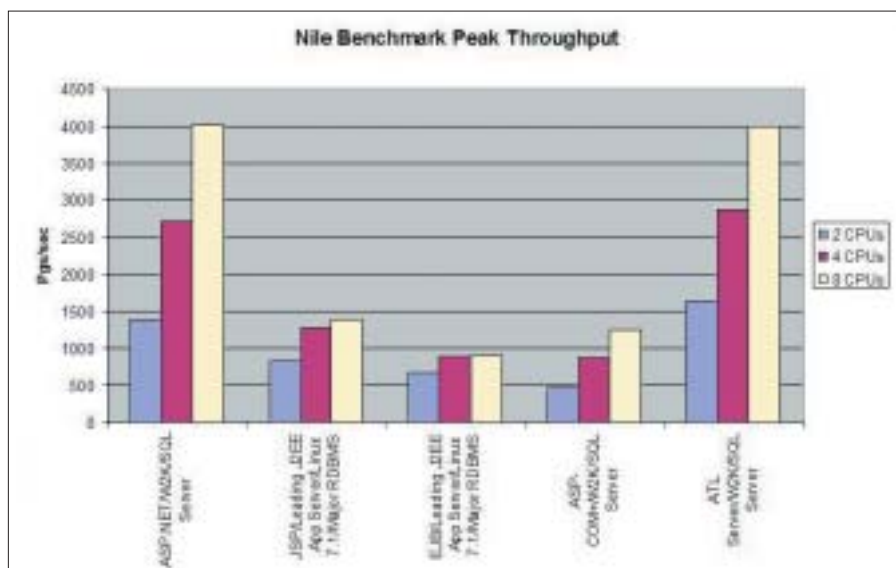
9. Foutafhandeling & Debugging

Alle talen in het .NET Framework hebben robuuste foutafhandeling. Daarnaast biedt ASP.NET de mogelijkheid tot het weergeven van een trace-log en debugging-informatie. Met Visual Studio.NET kun je bovendien code stap voor stap

uitvoeren. In web.config kun je verder standaardpagina's voor foutafhandeling opgeven.

10. Beveiliging

Naast de bestaande beveiliging van IIS, biedt ASP.NET nog enkele andere mogelijkheden. Verder werkt ASP.NET standaard niet in het SYSTEM-account, maar in het ASPNET-account. Dat account heeft beperkte toegang tot systeemresources, waardoor hacks op dat account minder problemen kunnen veroorzaken.



Afbeelding 5. Snelheidvergelijking tussen ASP.NET en andere platformen (bron: www.gotdotnet.com).

Nuttige internetadressen

- <http://www.asp.net>
- <http://www.dotnet.nl>
- <http://www.aspnl.com/aspnl/nl/lessen/>
- <http://www.aspnl.com/aspnl/nl/artikelen/>
- <http://samples.gotdotnet.com/quickstart/aspplus/>
- <http://www.gotdotnet.com/team/compare/nileperf.aspx>
- <http://www.123aspx.com>