



ASP.NET Community Starter Kit

BOUW JE EIGEN COMMUNITY SITE IN 5 MINUTEN

Stel je eens voor dat jij en een paar vrienden dezelfde interesse in oldtimers delen. Je hebt veel kennis op dit gebied vergaard en elke maand is er een bijeenkomst met een heerlijk diner gepaard aan een uitstapje met een mooie oldtimer. Maar eigenlijk wil je veel vaker dan slechts één keer per maand contact met jouw vrienden hebben. Je wilt ideeën uitwisselen en een gemeenschappelijke plaats hebben waar je kunt vertoeven. Je wilt de foto's van de laatste rit langs de drukke Amsterdamse grachten en een artikel over jouw favoriete auto delen, en ook wil je een discussie voeren over welke banden nu beter zijn voor jouw '64 Ford Mustang. Dit alles en nog veel meer kun je nu zelf doen op internet met hulp van ASP.NET Community Starter Kit.

Wat is de ASP.NET Community Starter Kit? Ten eerste is het een gemakkelijk te gebruiken gereedschap dat het mogelijk maakt een community website snel op te bouwen. Ten tweede is het, dankzij de open architectuur, een gereedschap dat je zelf kunt aanpassen, zodat het precies past bij jouw wensen en behoeftes. Hiermee kun je dan een community website creëren die er uit ziet zoals jij dat wilt en zich gedraagt zoals jij dat wilt.

Kant en klaar

Op het moment dat je de ASP.NET Community Starter Kit installeert is hij klaar om te gebruiken en heeft hij een voorbeeld geladen waardoor het gemakkelijk is meteen alle mogelijkheden te zien. Het startpunt van de website is de homepage. Op de homepage kun je verschillende secties hebben. Een sectie kan één van de volgende types zijn: article, books, events, links, photos, download, HTML page, discussion en parent. Het laatste type is een soort van groepering van andere sectietypes, waardoor het mogelijk is een soort structuur aan te brengen. Alle secties hebben hun eigen veiligheidsinstellingen. De Starter Kit biedt je

ook de mogelijkheid om, via een Web Service, je te abonneren op de inhoud van een andere Community Starter Kit. Via de Web Service kun je dan op gezette tijden de inhoud van de andere Starter Kit importeren. Dit is echter slechts een kort overzicht van de mogelijkheden van de Starter Kit. Op het moment dat je zelf met de Starter Kit gaat werken zul je een rijkdom aan functionaliteiten ontdekken.

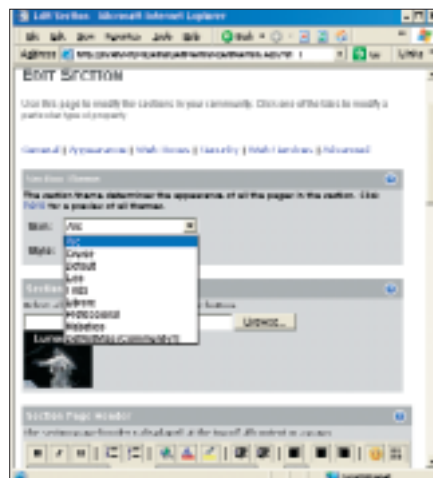
Kameleon

De ASP.NET Community Starter Kit is in feite een kameleon. Doordat je de vorm-

geving van de Starter Kit zo drastisch kunt veranderen vind ik dat het een tweede naam verdient: Kameleon. Om de vormgeving van de complete website te veranderen is het genoeg om naar het beheerdersgedeelte te gaan en gewoon een nieuwe skin en style te selecteren; zie afbeelding 1. Maar dat is nog niet alles. Je kunt ook voor elke sectie een andere skin en/of style kiezen. Een algemenere naam voor skin en style is theme.

Hoe werkt het?

Je hebt nu een algemeen beeld van wat de ASP.NET Community Starter Kit is. Daarom zullen we nu eens onder de 'motorkap' kijken, hoe het werkt en hoe we de Starter Kit kunnen wijzigen. De naam zelf zegt ons dat de Starter Kit is gebouwd met ASP.NET. De inhoud van alle secties, inclusief foto's en downloads en ook inclusief de website-structuur zijn in de database opgeslagen. Alle HTTP-verzoeken die binnen komen worden door CommunitiesModule HTTP Module afgehandeld en vervolgens doorgestuurd naar de communityDefault.aspx pagina, die in feite de enige aspx-pagina



Abbeelding 1. Selecteren van een skin of style

```

<%@ Control %>
<%@ Register TagPrefix="community" Namespace="ASPNET.StarterKit.Communities"
Assembly="ASPNET.StarterKit.Communities" %>
<%@ Register TagPrefix="links" Namespace="ASPNET.StarterKit.Communities.Links"
Assembly="ASPNET.StarterKit.Communities" %>
<%--

This skin is used for displaying an individual link in the Link page. Modify the
HTML content to modify the appearance of the page.
--%>

<div align="right">
    <community:DisplayTopic Runat="Server"/>
</div>
<table class="Link_Table">
<tr>
    <td class="Link_IntroCell">
        <community:LinkTitle CssClass="Link_Link" Runat="Server"/>
    </td>
</tr>
<tr>
    <td class="Link_BodyCell">
        <community:BriefDescription CssClass="Link_BriefDescription" Runat="Server"/>
    </td>
</tr>
</table>

<community:Rating id="ctlRateAction" Runat="Server"/>
<community:Comments id="ctlDisplayComments" Runat="Server"/>
<p>
<community:LinkEditContent CommentText="Add Your Comment"
    EditText="Edit this Link" DeleteText="Delete this Link" Runat="Server" />

```

Codevoorbeeld 1

op de website is. Wanneer je een Starter Kit-pagina aanroept, wordt een aantal skins geladen. Deze bepalen hoe de pagina er uit komt te zien. Als je bijvoorbeeld de Arc-skin hebt gekozen voor je website en je roept de Login-pagina aan, worden de volgende skins geladen. Dit is geen uitputtende lijst.

De Arc Page Skin gelegen in \Themes\Arc\Skins\PageSkins\Default.ascx

De Login pagina Content Skin gelegen in \Themes\Arc\Skins\ContentSkins\Users_Login.ascx

De Section Menu Control Skin gelegen in \Themes\Arc\Skins\ControlSkins\Sections_SectionMenu.ascx

Van belang is het feit dat skins kunnen overerven (inheritance). Als een skin van een bepaald thema niet kan worden

geladen, wordt die vanuit het Default-thema geladen. Dit is belangrijk omdat dit betekent dat je niet voor alle pagina's en controls skins hoeft aan te maken als je een nieuw thema wilt bouwen. Je kunt gewoon alleen de skins die je wilt wijzigen opnieuw schrijven.

Skins

Een skin is een ASP.NET user control die alleen HTML-code en ASP.NET-controls bevat, zonder de code-behind-pagina. Als voorbeeld kun je de Links_Link.ascx-skin voor de Link-pagina zien in code-voorbeeld 1. Er zijn vier types skins in de Starter Kit aanwezig: Page, Content, Control en Template. De Page-skin bepaalt de algemene lay-out van de pagina. De Page-skin bepaalt bijvoorbeeld de locatie van de sectie met het navigatiemenu. Normaal gesproken zou alleen

een Page-skin per website worden gekozen, die is gesitueerd in de PageSkins-map met de naam Default.ascx. De Content-skin is toegepast op het inhoudelijke gedeelte van de pagina. Er is bijvoorbeeld een Content-skin voor de Login-pagina die bepaalt hoe de pagina eruit ziet. Er is dan ook een één op één relatie tussen de Content-skin en pagina's op de website. De Control-skin is gebruikt om de lay-out van de control aan te passen. De sectie met de navigatiecontrol is bijvoorbeeld geïmplementeerd als een Control-skin. Dat betekent dat je de lay-out van het navigatiemenu kan veranderen door verschillende skins met de control te verbinden. Met de Template-skin bepaal je de verschijning van de controlsjabloon als ItemTemplate en HeaderTemplate in de Repeater-control.

```

namespace ASPNET.StarterKit.Communities.Cars {
    using System;
    using System.Web.UI;
    using System.Web.UI.WebControls;
    using System.Web.UI.HtmlControls;
    using ASPNET.StarterKit.Communities;

    //*****
    // EditCar Class
    // Represents the Edit Car page. Enables users to edit cars.
    //*****
    public class EditCar : ContentEditPage {
        string _skinFileName = "Cars_AddCar.ascx";
        string _sectionContent = "ASPNET.StarterKit.Communities.Cars.CarSection";
        TextBox txtCarName;
        TopicPicker dropTopics;
        TextBox txtCarCompany;
        TextBox txtCarPrice;
        TextBox txtCarPurchaseLink;
        TextBox txtCarBriefDescription;
        HtmlTextBox txtCarFullDescription;
        HtmlInputFile txtImageFile;
        DisplayCarImage imgCarImage;
        TextBox txtCarManufactureDate;
        TextBox txtCarCountry;
        TextBox txtCarCompanyLink;

        //*****
        // ContentPageID Property
        // Stores the ID of the content being edited in View State
        //*****
        int ContentPageID {
            get { return (int)ViewState["ContentPageID"]; }
            set { ViewState["ContentPageID"] = value; }
        }

        //*****
        // ImageID Property
        // Represents the Image ID of the car being edited.
        //*****
        int ImageID {
            get { return (int)ViewState["ImageID"]; }
            set { ViewState["ImageID"] = value; }
        }

        //*****
        // SkinLoadCar
        // The skin load event happens after a page skin has been loaded.
        // Here, we grab the necessary controls from the page skin.
        //*****
        void SkinLoadCar(Object s, SkinLoadEventArgs e) {
            // Find the Form so that we can change the EncType of the post.

            HtmlForm form = (HtmlForm)Page.FindControl("PageForm");
            form.Enctype="multipart/form-data";

            // Find Car Title
            txtCarName = (TextBox)GetControl(e.Skin, "txtCarName" );

```

(Zie vervolg code op volgende pagina)

(vervolg code vorige pagina)

```
// Add here the rest of the controls...
}

//*****
// OnLoad Method
// Assign previous content values to edit form.
//*****
override protected void OnLoad(EventArgs e) {
    string imagePath;

    if (!Page.IsPostBack) {
        // Get the ContentPageID from QueryString
        ContentPageID = Int32.Parse(Page.Request.QueryString["id"]);

        // Get the car info
        CarInfo _carInfo = (CarInfo)CarUtility.GetCarInfo(objUserInfo.Username, ContentPageID);

        // Store the original Image ID
        ImageID = _carInfo.ImageID;

        // Populate the controls
        EnsureChildControls();

        // Assign value to topics
        dropTopics.SelectedTopicID = _carInfo.TopicID;

        txtCarName.Text = _carInfo.Title;
        txtCarCompany.Text = _carInfo.CarAuthor;
        txtCarPurchaseLink.Text = _carInfo.PurchaseLink;
        txtCarPrice.Text = _carInfo.Price.ToString("n2");
        txtCarBriefDescription.Text = _carInfo.BriefDescription;
        txtCarFullDescription.Text = _carInfo.FullDescription;
        txtCarManufactureDate.Text = _carInfo.PublicationDate.ToString("d");
        txtCarCountry.Text = _carInfo.Publisher;
        txtCarCompanyLink.Text = _carInfo.PublisherLink;
        if (_carInfo.ImageID != -1) {
            imgCarImage.Visible = true;
            imagePath = ImageUtility.BuildImagePath(_carInfo.ImageID, _carInfo.ImageName);
            imgCarImage.ImageUrl = CommunityGlobals.CalculatePath(imagePath);
        }
        else
            imgCarImage.Visible = false;
    }
}

//*****
// SubmitCar Method
// This method is raised by clicking the Edit button in the Edit form.
//*****
private void SubmitCar(Object s, EventArgs e) {
    // Add here code for saving the content from the page into the DB...
}

//*****
// EditCar Constructor
// Calls the base SkinnedCommunityControl constructor and assigns the default page skin.
// Also checks whether current user has permissions to edit.
//***** (Zie vervolg code op volgende pagina)
```

(vervolg code vorige pagina)

```
public EditCar() : base() {
    // Assign a default skin file name
    SkinFileName = _skinFileName;

    // Specify Section Content
    SectionContent = _sectionContent;

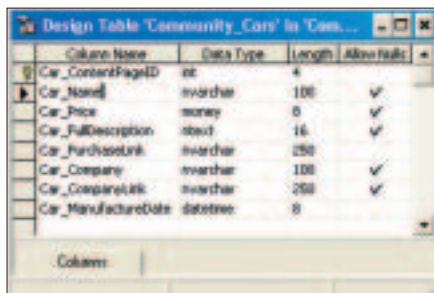
    // Wire-up event handlers
    this.SkinLoad += new SkinLoadEventHandler(SkinLoadCar);
    this.Submit += new SubmitEventHandler(SubmitCar);
}
}
```

Codevoorbeeld 2.

Alle pagina's in de Community Starter Kit zijn geïmplementeerd als classes die uiteindelijk erven van de SkinnedCommunityControl basisclass. De SkinnedCommunityControl basisclass zorgt voor alle details over het laden van de juiste skin voor een control. De class heeft een belangrijke property en methode die je moet overschrijven: de SkinType-property en de InitializeSkin-methode. De SkinType-property geeft het type van de skin aan die is gekoppeld met de control: ContentSkin, ControlsSkin of TemplateSkin. De InitializeSkin-methode haalt alle controls op uit de skin, zodat de property's van de controls programatisch kunnen worden aangepast. Bijvoorbeeld, in de InitializeSkin-methode haal je een button-control op en geef je voor deze button een OnClick-event aan.

Modules

Zoals eerder genoemd is er een aantal kant en klare modules (sectie types) die je kunt gebruiken om een nieuwe sectie in je website aan te maken. Als je niet blij bent met bepaalde functionaliteit of je wilt iets nieuws toevoegen, zoals een automodule, heb je twee opties. De eerste optie is om de huidige module aan



Column Name	Data Type	Length	Allow Nulls
Car_ContentPageID	int	4	
Car_Name	nvarchar	100	✓
Car_Price	money	8	✓
Car_FullDescription	ntext	16	✓
Car_HandbookLink	nvarchar	250	✓
Car_Company	nvarchar	100	✓
Car_CompanyLink	nvarchar	250	✓
Car_ManufactureDate	datetime	8	

Afbeelding 2. Aanmaken Community_Cars-tabel

te passen en de tweede is een nieuwe module in de Starter Kit aan te maken. Wat zijn nu de nodige stappen om de tweede optie te bereiken?

Het eerste wat je moet doen is een nieuwe tabel in de CommunityStarterKitDB-database aanmaken. Deze tabel wordt gebruikt om gegevens (de inhoud) van onze nieuwe module te bewaren. Maak bijvoorbeeld een Community_Cars-tabel aan; zie afbeelding 2.

De tweede stap is om een stored procedure in de database te schrijven. De procedure wordt gebruikt voor het invoegen, bewerken en verwijderen van gegevens uit de database. Deze stap is optioneel, maar als je besluit die niet te doen dan moet je deze databaseacties zelf volledig programmeren in je code.

Ten derde moet je een aantal classes creëren die erven van de basisclass SkinnedCommunityControl. Deze classes vertegenwoordigen de pagina's in de nieuwe module. Je zou hoogstwaarschijnlijk de volgende pagina's willen hebben: Car Section-pagina, Car-pagina, Add Car en Edit Car-pagina. Om deze pagina's te creëren moet je vier classes schrijven. De Starter Kit maakt dat gemakkelijk. Er zijn vier basisclasses die je kunt gebruiken. Voor ieder paginatype is er een basisclass: ContentListPage, ContentItemPage, ContentAddPage en ContentEditPage; zie codevoorbeeld 2.

De vervolgstap is om nieuwe skins in de ContentSkins-map in het Default-thema toe te voegen: Cars_CarSection.ascx, Cars_Car.ascx en Cars_AddCar.ascx.

Let op, er zijn alleen drie skins! Alle standaardmodules in de Starter Kit gebruiken dezelfde skin voor Add- en Edit-pagina's. Vergeet in deze stap niet om nieuwe noodzakelijke classes toe te voegen aan de Default.css style (cascading stylesheet) van nieuwe pagina's waarvan je gebruik gaat maken. Voor een skin is het mogelijk verscheidene styles te hebben, door meer css-bestanden in de stylemap te hebben. Met behulp van styles het is ook mogelijk de verschijning van de pagina te veranderen, maar de veranderingen zijn niet zo extreem als je kan dat doen met skins. Na de voltooiing van de vorige stap moet je de Starter Kit opnieuw compileren. Indien er geen foutmeldingen zijn, is het enige dat nog rest de nieuwe module te registreren. Om te registreren dienen in beide databasetabellen Community_NamedPages en Community_PageTypes de nieuwe gegevens worden toegevoegd. Na de registratie verschijnt de nieuwe automodule in de lijst van beschikbare modules. Het eindresultaat zal er ongeveer als in afbeelding 3 uitzien.

Conclusie

Het is duidelijk hoe gemakkelijk het is de Starter Kit gewoon te installeren en deze te gebruiken. Ook is duidelijk hoe gemakkelijk het is de bestaande functionaliteit en lay-out te veranderen en uit te breiden, of nieuwe toe te voegen. Dit gereedschap is niet alleen voor ontwikkelaars bedoeld, maar ook voor gewone gebruikers en ISP's. Een interessant aspect voor ISP's is de

mogelijkheid om verschillende community websites te hebben met één enkele installatie van de Starter Kit. Er zijn al verscheidene ISP's die een automatische plaatsing van ASP.NET Starter Kits ondersteunen. De Community Starter Kit is in een C#, VB.NET- en J#-versie verkrijgbaar. Ook is een speciale 'command line' installatieversie voor ISP's. Mijn motto wat betreft de ASP.NET Community Starter Kit: "Het is wat je ervan maakt!". Veel plezier ermee!



Afbeelding 3. Eindresultaat nieuwe automodule

Nuttige Internetadressen

<http://www.asp.net/StarterKit>
<http://www.CommunityStarterKit.com>
<http://www.microsoft.com/vstudio>
<http://www.microsoft.com/netherlands/msdn/aspnet>
<http://msdn.microsoft.com/asp>

ASP.NET Starter Kit

De ASP.NET Starter Kits vormen de perfecte manier om kennis te maken met alle ASP.NET-technologieën voor het ontwerpen van totaaloplossingen. De onderdelen die aan bod komen grijpen alle terug op een aantal ASP.NET-technologieën en laten zien hoe u hiermee een praktische oplossing bouwt. De ASP.NET starter kits zijn bovendien gratis.

Beginners en gevorderden

Elke Starter Kit heeft een eigen moeilijkheidsgraad en vereist een andere instapniveau. Het is dan ook belangrijk dat een ontwikkelaar die voor het eerst met ASP.NET aan de slag gaat, begint met de minder ingewikkelde toepassingen. De Commerce Starter Kit is bestemd voor beginners en de Portal Starter Kit voor gevorderden. Het staat u namelijk volledig vrij om alle code in de voorbeelden opnieuw te gebruiken.

Features

Iedereen die met een ASP.NET Starter Kit aan het experimenteren slaat, ongeacht het instapniveau, moet weten welke features in een toepassing aanwezig zijn. Als u dan een bepaalde feature of technologie op het oog hebt, kunt u direct zien welke toepassing daar bij hoort. In de tabel hieronder hebben we alle features van de demotoepassingen in de twee starter kits op een rijtje gezet.

Feature	Commerce	Portal	Time Tracker	Reports	Community
Features voor toepassingen					
Beveiligde toegang	●	●	●	●	●
Beveiligde toegang op basis van rollen	○	●	●	○	●
Dynamisch gegenereerde UI	○	●	○	○	●
Winkelwagen	●	○	○	○	○
Catalogus	●	○	○	●	○
Portal	○	●	○	○	●
Discussielijst	○	●	○	○	●
Custom error handling	●	●	●	●	○
On line beheer	○	●	●	○	●
On line content bewerken	○	●	●	○	●
Content Rating	○	○	○	○	●
Web Farmable	●	●	●	●	●
Orderinvoer en -status B2B	●	○	○	○	○
Pagina's					
Cross-browser support	●	●	●	●	●
Dynamische UI aanmaken	○	●	○	○	●
Server Controls	●	●	●	●	●
User Controls	●	●	●	●	●
Custom Controls	○	● (mobiel)	○	○	●
Mobiele gebruikersinterface	○	●	●	○	○
Gegevens					
App Settings	●	●	●	●	●
Stored Procedures	●	●	●	●	●
Databasetransacties	●	○	●	○	○
DataReaders	●	●	○	○	●
DataSets	●	●	●	●	●
Beveiliging					
Forms-gebaseerde cookieverificatie	●	●	●	○	●
Windows NTLM Security	○	●	●	○	○
Beveiliging op basis van rollen	○	●	●	○	●
Caching					
Output Caching	●	●	○	○	●
Fragment Caching	●	●	○	○	●
Data Caching	○	●	○	○	●
Context.Items	○	●	○	○	●
Diversen					
Webservices	●	○	○	○	●
Mobile Support	○	●	●	○	○

Tabel 1. Overzicht technologieën in ASP.NET Starter Kits

Installatievereisten

Voor het installeren en gebruiken van de ASP.NET Starter Kit zijn er de volgende vereisten:

- * Microsoft ASP. Microsoft .NET Framework
- * SQL Server of Microsoft Data Engine (MSDE) versie 7.0 (of later)
- Opmerking: Voor de ASP.NET Portal Starter Kit hebt u ook nog nodig:
 - SQL Server 2000 of MSDE 2000 (of later)
 - Microsoft Mobile Internet Toolkit
- * Windows 2000 (Professional, Server of Advanced Server) of Windows XP Professional

Wat zit er in de kit?

Het installatiepakket, dat op de ASP.NET website staat, bevat een complete installatietoepassing evenals de toepassing zelf, de broncode en relevante documentatie. Elk installatiepakket bevat het volgende:

- * Volledige broncode (in C# of Visual Basic.NET)
- * Broncodeviewer
- * Documentatie
- * Setuppakket voor HTA (HTML-toepassing)