

Inloggen met ASP.NET 2.0

BOUW JE EIGEN LOGIN MET DE VERZAMELING CONTROLS IN ASP.NET 2.0

In ASP.NET 2.0 is veel aandacht besteed aan het inloggen en registreren van gebruikers op de website. Met behulp van diverse standaardcontrols, een database en een administratie-tool is het maken van een login-module eenvoudig. Er hoeft (bijna) geen code geschreven te worden om het geheel werkend te krijgen. Mits de gebruiker niet veeleisend is. Want hoe meer aanpassingen aan de standaardmodule, hoe groter de complexiteit. In dit artikel beschrijft de auteur niet alleen de ASP.NET-configuratie, maar gaat hij ook in op de werking van het login-control en de Create User Wizard. Bovendien legt hij uit hoe een gebruiker zijn wachtwoord kan wijzigen met het changepassword-control.

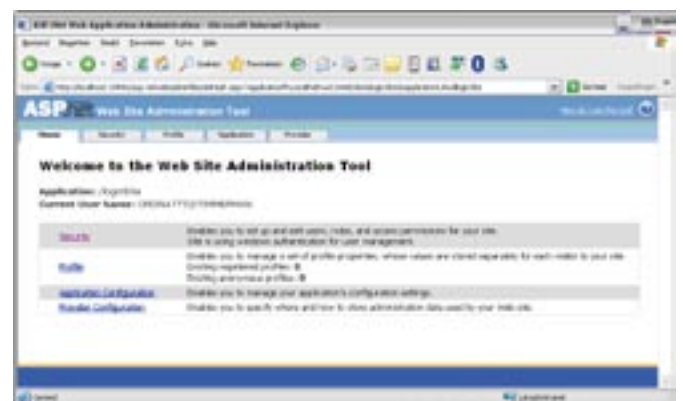
Met de komst van ASP.NET 2.0 is het aantal kant- en klare controls aanzienlijk uitgebreid. Ook de mogelijkheden die deze standaard controls nu bieden is vergroot. De login-module bestaat uit een aantal onderdelen:

- ASP.NET-configuratie-tool. Een tool om een database-provider te kiezen, gebruikers en gebruikersrollen te beheren en diverse webconfiguratie-settings te beheren.
- Een database. De database-provider kan een Microsoft Access-database of een SQL Server-database zijn waar de gebruikeraccountgegevens worden opgeslagen. Het beheer van de provider kun je via de ASP.NET website-administration-tool regelen.
- De webconfiguratie. In het 'web.config'-bestand wordt de link gelegd tussen de diverse login-controls en de databaseproviders. Het beheren van diverse configuratie-instellingen kan nu ook worden gedaan via de ASP.NET website-administration-tool.
- Het login-control. Een User Interface (UI) template based control bedoeld voor het inloggen van de gebruiker.
- Het login name-control. Een control die de naam van de ingelogde gebruiker weergeeft.
- Het login status-control. Een control die aangeeft of de gebruiker ingelogd is.
- Het login view-control. Een template-based control waarbij per rol de template aangepast kan worden en waarin per gebruikersrol specifieke HTML geplaatst kan worden. De gebruiker ziet daardoor alleen de informatie die aan de gebruikersrol is gekoppeld.
- De CreateUserWizard. Een template-based control voor het aanmaken van een gebruikersaccount.
- De ChangePasswordControl. Een template-based control voor het veranderen van een wachtwoord.
- De PasswordRecoveryControl. Een template-based control voor het versturen van het wachtwoord naar de gebruiker.
- De Membership(provider)-klassen. Dit is de business-klasse achter de controls. Het is mede een klasse waarmee gegevens over gebruikers opgevraagd kunnen worden en nieuwe wachtwoorden automatisch kunnen worden gegenereerd.

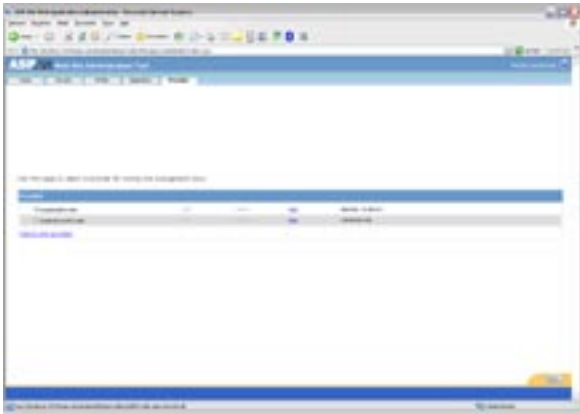
Dit zijn nogal wat onderdelen. Het voordeel van de controls is dat ze samenwerken en eenvoudig aan te passen zijn. Een nadeel is dat het beheer van gebruikers min of meer gekoppeld wordt aan Microsoft Access of SQL Server. Hierdoor wordt het moeilijker de module aan een ander type database te koppelen. Ook wordt het minder gemakkelijk gemaakt een n-tier-applicatie te gebruiken. De enige manier om deze opties te gebruiken is door zelf een membershipprovider te maken. Deze provider moet aan een bepaalde interface voldoen, namelijk die van de MembershipProvider-klasse. Hoewel het maken van een dergelijke provider goed te doen is, bevat deze abstracte klasse veel methodes die geïmplementeerd moeten worden. Een ander nadeel van een eigen provider is dat de website administration-tool voor het beheren van gebruikers en rollen met deze custom provider niet gebruikt kan worden.



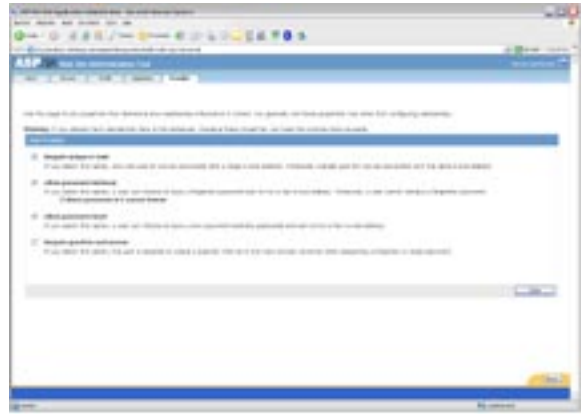
Afbeelding 1. Het login-control



Afbeelding 2. Het openingsscherm van de Website Administration Tool



Afbeelding 3. Zelf een provider toevoegen

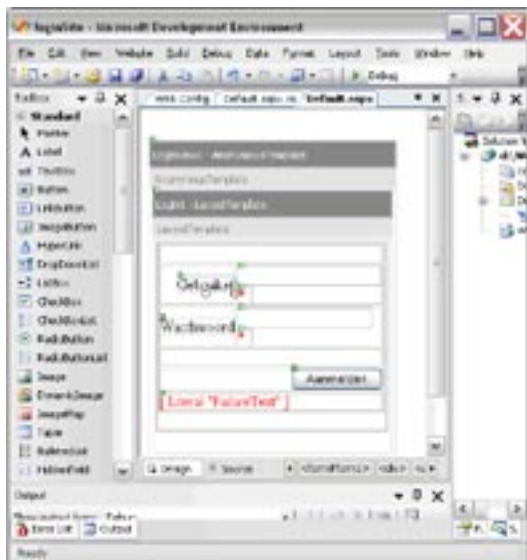


Afbeelding 4. Selecteren van accountinstellingen

De ASP.NET-configuratie

De ASP.NET-configuratie kan worden gedaan met de Web Site Administration Tool. Deze tool bewerkt de web.config. Er kunnen applicatiesettings mee worden beheerd en deze worden aangepast in de web.config-file. Ook kan een provider worden toegevoegd. Een provider is in dit geval een databaseconnectie. Deze connectie wordt mede gebruikt om een aantal standaard tabellen aan te maken. Deze tabellen bewaren de gegevens van gebruikers. Via de tool kunnen ook gebruikersrollen worden aangemaakt. Gebruikers kunnen aan rollen worden toegevoegd. Aangezien het aanmaken van rollen vaak inhoudt dat een gebruiker een bepaald deel van de site wel of niet mag zien, is het beheren van deze mogelijkheid ook opgenomen. Zo is in te stellen dat alleen de beheerder van de site bij de beheerpagina's kan komen. Komt een gebruiker, die geen beheerder is, toch op de URL van een beheerderpagina, dan kan de gebruiker naar een andere pagina worden geleid. Dit concept is ook van toepassing op anonieme webgebruikers. Het login-control kan gemakkelijk op het scherm worden geplaatst en ziet er uit zoals in afbeelding 1.

Het control heeft één lay-out-template. Om deze te kunnen bewerken moet het control eerst naar een templated control geconverteerd worden. Daarna kan de lay-out van de template worden aangepast. Het is mogelijk extra velden toe te voegen en het item 'remember me next time' te verwijderen. Echter, wat zit achter deze control, aangezien het programmeren van C#-code overbodig is? Om het control te laten werken is meer nodig. Via het menu 'website' kan de 'ASP.NET configuration' worden gestart; zie afbeelding 2. Deze persisteert diverse gebruikersinformatie met een database.



Afbeelding 5. Het login view-control en het login-control in design

De tool zelf is webbased en er zit een time-out op van een minuut. Dus na een minuut idle te zijn geweest dien je de tool opnieuw op te starten. Om de gebruikersgegevens op te slaan moet eerst een provider worden gekozen. Standaard is dit de Microsoft Access-provider. Ook is het mogelijk om voor diverse configuratiedoelinden een andere provider te kiezen. De keuze bestaat voornamelijk tussen Microsoft Access en SQL-serverproviders. Mijn voorkeur gaat uit naar SQL Server, omdat deze schaalbaarder is. Als je niet in het bezit bent van SQL Server kan je ook SQL Server Express gebruiken. Ik voeg als eerste stap een provider toe, zodat ik op een website in kan loggen; zie afbeelding 3.

In het geval van SQL Server dient een gebruikersvriendelijke naam en een fysieke databasenaam te worden ingevuld. Als gebruik wordt gemaakt van een andere server of van de authenticatiemodus dienen deze gegevens ook ingevuld te worden. Deze gegevens vormen samen de databaseconnectie. Vervolgens maken we een aantal keuzes om de membership-gegevens te bewaren. De laatste vraag van de keuzes zet ik liever uit. Het idee jezelf een vraag te stellen en een verborgen antwoord daarop te krijgen, lijkt me onzinnig. Te veel gedoe om je vergeten wachtwoord terug te halen. Het idee hierachter is om het vergeten wachtwoord naar het e-mailadres op te sturen. Met een vraag-en-antwoordspel wordt dit voor de hacker en de gebruiker minder gemakkelijk gemaakt. Een andere optie is om een nieuw wachtwoord te genereren voor de persoon in kwestie. In de bètaversie blijkt het nog niet mogelijk de encryptie van het wachtwoord aan te zetten en daarna een gebruiker aan te maken. Dus dit vinkje zet ik niet aan. Het wachtwoord wordt echter wel beveiligd opgeslagen; zie afbeelding 4. Vervolgens test ik de gemaakte databaseprovider. Indien alles lukt, kan er via het tabblad 'security' users, rollen, koppelingen van users aan rollen en rollen aan websites-directories worden gemaakt. Ik bedenken een rol-user en voeg deze toe. Daarnaast kunnen er op het tabblad 'profile' nog extra eigenschappen worden gedefinieerd. Deze eigenschappen koppelen je aan de gebruiker en vormen het profiel van een gebruiker. In het tabblad 'applicatie' kunnen we applicatievariabelen opnemen, de custom-foutpagina zetten, smtp-gegevens wijzigen en sitecounter-informatie configureren. Alle settings worden in het web.config-bestand bewaard. Alle data worden opgeslagen in de database van de gekozen provider(s).

Het login-control

Het login-control is eenvoudig. Er zijn drie velden, namelijk login, wachtwoord en onthouden ja/nee. De lay-out laat te wensen over. Het is aan te raden de lay-out aan te passen en van het control een templated-versie te maken. Het veld 'remember me' kan eventueel zonder problemen worden verwijderd. Als in de ASP.NET configuratie-tool de provider is aangemaakt en ten minste een user is aangemaakt, kan de werking van het login-control worden getest; zie afbeeldingen 5 en 6. Dat er geen codevoorbeeld is, zegt genoeg over de kant die de applicatieontwikkeling opgaat. Het is meer



Afbeelding 6. Het login view-control en het login-control in gebruik

een kwestie van configureren. Ik ga nu testen hoe het inloggen verloopt met behulp van een loginview-control. Allereerst sleep ik een loginview-control van de toolbox op het formulier. Ik zet de loginview-controltemplate op anonymous login. Deze lay-out krijgt de gebruiker te zien als hij zich nog niet heeft aangemeld. Daarna sleep ik het login-control, dat ik al op het formulier heb geplaatst, op de template van het loginview-control. Ik zet de loginview-template vervolgens op 'loggedInTemplate'. Deze template krijgt de gebruiker te zien nadat hij succesvol heeft ingelogd. Ik sleep het loginname-control en het loginstatus-control vanaf de toolbox in de template van het loginview-control. Het loginname-control zal daardoor na het inloggen de naam van de gebruiker tonen en de loginstatus maakt het mogelijk dat de gebruiker weer kan uitloggen. Ik test het project en voilà, ook dit werkt zonder het schrijven van één regel programmacode.

De Create User Wizard

Met behulp van het CreateUserWizard-control kan een nieuwe gebruiker zich laten registreren. Dit gebeurt meestal op een andere pagina dan de loginpagina. Om de link met deze nieuwe pagina te maken, sleep ik een hyperlink-control in de anonymous-template van het loginview-control en laat het href-attribuut verwijzen naar de pagina 'register.aspx'. Ik voeg een nieuwe webform-pagina vervolgens toe aan de solution. Daarbij kies ik voor de optie met bestand achter de code. Dat de code tegenwoordig in de aspx geprogrammeerd kan worden, vind ik zelf niet zo'n verleidelijk alternatief. Ik sleep vervolgens het CreateUserWizard-control op het formulier. Het control heeft vele eigenschappen om teksten van labels en foutboodschappen te zetten. Het is mogelijk om extra stappen toe te voegen aan de wizard. Dit maakt het werken met het control gecompliceerder. De CreateUserWizard is afgeleid van de Wizard-control en bevat diverse templates.

Ik heb ervoor gekozen om createuser-navigatietemplates te gaan bewerken en een nieuwe template toe te voegen. Op de navigatie-templates komen minimaal de verplichte buttons die afgeleid zijn van IButtonBase. De ID's van de buttons hebben een verplichte vaste naam. Je zou verwachten dat door deze verplichting de buttons direct werken, maar dat is niet het geval. De 'continue'-button van de navigatie-template die bij de 'Create useraccount' hoort, is bedoeld om een gebruiker toe te voegen waarna de volgende stap van de wizard volgt. De 'continue'-button dient door aanpassing handmatig geïmplementeerd te worden. Daarbij valt gelijk op dat de controls van de 'Create User Account'-template niet benaderbaar zijn. Het login-control heeft gelukkig standaardeigenschappen zoals 'UserName' en 'Password' en 'Email'. Wil je zelf een veld op een template toevoegen, dan is deze op de eerste pagina van de

```
switch (CreateUserWizard1.ActiveStepIndex)
{
    case 0: //first page create account
        string userName = Server.HtmlEncode(CreateUserWizard1.UserName);
        string password = Server.HtmlEncode(CreateUserWizard1.Password);
        string email = Server.HtmlEncode(CreateUserWizard1.Email);
        MembershipCreateStatus result;
        MembershipUser user;
        user = Membership.CreateUser(userName, password, email, null,
            null, true, out result);
        switch (result)
        {
            case MembershipCreateStatus.Success:
                Roles.AddUserToRole(user.UserName, "user");
                CreateUserWizard1.ActiveStepIndex += 1;
                break;
        }
        break;
}
```

Codevoorbeeld 1.

wizard moeilijk te benaderen. Dit geldt niet voor de toegevoegde pagina's. Gek genoeg zijn de velden die op templates van een andere pagina zijn geplaatst wel benaderbaar. Dus wil je voor het creëren van een account bij de gebruiker extra gegevens invullen, voeg dan een WizardStep toe.

Om mijn button te laten werken implementeer ik de code als in codevoorbeeld 1 onder de 'continue'-button.

De velden naam, wachtwoord en e-mail zijn standaardeigenschappen van het control. Het membership-object zorgt voor het beheer van gebruikers. Door de 'CreateUser'-methode aan te roepen wordt een gebruiker aangemaakt. De methode kent een output-parameter. Deze output-parameter geeft de status van het aanmaken aan. De methode zelf retourneert een gebruiker. Wanneer de output is geslaagd, voeg ik een rol aan de gebruiker toe. Ten slotte geef ik de wizard de opdracht om naar de volgende pagina te gaan door de eigenschap 'ActiveStepIndex' met 1 op te hogen.

Ik voeg een WizardStep toe aan het loginview-control. Daarop plaats ik een label en een textbox. De label krijgt de tekst geboortedatum en de textbox krijgt de naam 'txtGeboorteDatum'. Daarna voeg ik een profielgroep 'User' toe en koppel hieraan een eigenschap 'GeboorteDatum' van het type DateTime. Ik gebruik hiervoor de website administration-tool; zie afbeelding 7.

Ik test het loginview-control. Ik merk dat de gebruiker wordt aangemaakt en dat ik alles kan invullen bij geboortedatum. Ik besluit een custom validator in te vullen. De code die ik achter de validator.ServerValidation-event invul is te zien in codevoorbeeld 2.

Hiermee controleer ik of de datum voldoet aan het formaat. Indien dit niet het geval is, mislukt de parse-functie en wordt de foutboodschap getoond. De code onder de 'continue'-button vul ik aan met codevoorbeeld 3.

Eerst wordt het profiel opgehaald van de gebruiker. Ik vul op het

```
try
{
    DateTime.Parse(this.txtGeboorteDatum.Text);
}
catch (Exception ex)
{
    args.IsValid = false;
    this.CustomValidator1.Text = ex.Message;
}
```

Codevoorbeeld 2.

```

case 1: //second page extra info
HttpProfile p = Profile.GetProfile(CreateUserWizard1.UserName);
p.User.GeboorteDatum = Convert.ToDateTime(txtGeboorteDatum.Text);
p.Save();
break;

```

Codevoorbeeld 3.

```

default:
FormsAuthentication.Authenticate
(CreateUserWizard1.UserName, CreateUserWizard1.Password);
break;

```

Codevoorbeeld 4.



Afbeelding 7. Een eigenschap aan een gebruikersprofiel toevoegen

profiel van de door mij aangemaakte profielgroep user de eigenschap geboortedatum in. Vervolgens worden de gegevens via de save-methode bewaard. Om een net aangemaakte gebruiker automatisch in te loggen via forms-authentication voer ik het volgende statement in aan het eind van het wizardproces. Dit kan onder meer worden gedaan onder de default-optie van de codevoorbeeld 4.

Het loginview-control kent standaard eigenschappen om te navigeren. Wanneer de eigenschap 'ContinueDestinationPageURL' wordt ingevuld, ga je na de eerste pagina naar deze webpagina. Wanneer de wizard wordt geannuleerd, wil je meestal terug naar de pagina waar je de wizard hebt opgestart. Deze pagina-URL kan in de 'CancelDestiantionPageURL'-eigenschap ingevuld worden. Met de eigenschap 'FinishDestinationPageURL' kan na het doorlopen van de wizard de juiste pagina worden getoond.

Change password and Password recovery

Met het change password-control kan de gebruiker zijn wachtwoord wijzigen. Het password recovery-control stelt een gebruiker in staat zijn wachtwoord op te vragen. Het 'change password-control' is gemakkelijk te gebruiken. De belangrijke eigenschappen van het control zijn te zien in tabel 1.

Met deze kennis sleep ik een 'ChangePassword'-control naar een nieuwe webpagina. Op de login-pagina voeg ik in de 'loggedIn-Template' van het loginview-control een hyperlink toe die naar deze nieuwe pagina verwijst. Ik vul als control-eigenschappen bij AuthenticationFailureAction de waarde 'refresh' in en bij succes

Eigenschap	Opmerking
AuthenticationFailureText	Foutboodschap indien het wijzigen van het wachtwoord mislukt.
AuthenticationFailureAction	Keuze: opnieuw of naar loginpagina. Indien vernieuwen ('refresh') krijgt de gebruiker nog een kans om het wachtwoord te wijzigen.
CreateUserText	Tekst bij een hyperlink. Deze hyperlink verwijst naar een pagina waar een gebruiker zich kan registreren.
CreateUserURL	Webpagina waar een gebruikersaccount kan worden aangemaakt.
SuccessPageURL	Indien het wijzigen van het wachtwoord is geslaagd, ga je naar deze webpagina.
SuccessText	Tekst bij een geslaagde wijziging van het wachtwoord.
DisplayUserName	Als 'true' dan wordt de gebruikersnaam van de ingelogde gebruiker getoond in het control.
CancelDestinationPageURL	Pagina waarnaar de gebruiker wordt gestuurd als hij zijn wachtwoord niet wil wijzigen.

Tabel 1. Eigenschappen van het password-control

verwijs ik weer naar de loginpagina. De DisplayUserName zet ik op 'true', zodat ik kan zien onder welke gebruikersnaam ik ben ingelogd. Alle eigenschappen met Engelse teksten verander ik in Nederlandse teksten. Bij de CancelDestinationPageURL vul ik ook de loginpagina in. Het lukt me het control te testen en een wachtwoord te veranderen.

Voor de password-recovery sleep ik het 'passwordRecovery'-control van de toolbox op een nieuwe webpagina. Ik merk al snel dat het control gebruikmaakt van mail. Hoewel ook andere login-controls dit aspect kennen is het voor dit control essentieel. De eigenschap 'Maildefinition' is onderverdeeld in diverse mail-eigenschappen. Om de mail te laten werken, pas ik in de website-administration-tool onder het tabblad 'application setting' de smtp-gegevens aan. Ik geef aan wat de mailserver van mijn provider is en vul het 'from address' in. Dit is het e-mailadres vanwaar ik berichten stuur. In de maildefinition vul ik een subject in. Ik voeg in de 'anonymous'-template van de login-pagina een hyperlink toe naar de nieuwe pagina. Ik test de password recovery-pagina. Ja, de verstuurd mail aan mezelf bevat de accountgegevens. Maar hoe ga ik dit bericht aanpassen aan mijn wensen? Ik kan alleen een tekstbestand toevoegen. Als ik dat doe, gaan de belangrijkste gegevens met accountinformatie uit het bericht verloren. Er is nog een ander probleem. Als ik de pagina 'successPageURL' invoer, ga ik direct na het opvragen naar een volgende pagina toe. Als ik dit niet doe krijg ik alleen de tekst van eigenschap 'SuccessText' te zien. Ik besluit alleen de tekst in te voeren, maar liever zou ik de gebruiker een melding geven met daarna de mogelijkheid om naar de beginpagina van de website te gaan.

Veel werk verzet, maar...

Microsoft heeft heel wat werk verzet om het inloggen en beheren van accounts op een website gemakkelijker te maken. Hierbij is het motto: 'met zo min mogelijk programmacode'. Ik geef ze daarin gelijk. De controls zijn gemakkelijk te gebruiken en in combinatie met de website-administrationtoolkit is het ook gemakkelijk om accounts te beheren. Nadelen zijn er echter ook. Met name op het moment dat het control niet voldoet, kan het lastig worden. Zo worden n-tier-applicaties niet ondersteund en kan het mailbericht van de passwordrecovery-control niet (of niet gemakkelijk) worden gewijzigd.

Dieder Timmerman werkt als Senior Software Engineer bij Ordina (<http://www.ordina.nl>). Hij is MCSD gecertificeerd en legt zich voornamelijk toe op het ontwikkelen van .NET-applicaties met behulp van ontwikkelstrategieën. Voor vragen en opmerkingen is hij te bereiken via dieder.timmerman@ordina.nl

Nuttige internetadressen

<http://msdn.microsoft.com/asp.net/>
http://msdn.microsoft.com/library/en-us/dnvs05/html/Person_fin.asp
<http://www.asp.net/>