Introduction to Membership

.NET Framework 4 59 out of 88 rated this helpful

ASP.NET membership gives you a built-in way to validate and store user credentials. ASP.NET membership therefore helps you manage user authentication in your Web sites. You can use ASP.NET membership with ASP.NET forms authentication by using with the ASP.NET login controls to create a complete system for authenticating users.

ASP.NET membership supports facilities for:

- Creating new users and passwords.
- Storing membership information (user names, passwords, and supporting data) in Microsoft SQL Server, Active Directory, or an alternative data store.
- Authenticating users who visit your site. You can authenticate users programmatically, or you can use the ASP.NET login controls to create a complete authentication system that requires little or no code.
- Managing passwords, which includes creating, changing, and resetting them . Depending on membership options you choose, the membership system can also provide an automated password-reset system that takes a user-supplied question and response.
- Exposing a unique identification for authenticated users that you can use in your own applications and that also integrates with the ASP.NET personalization and role-management (authorization) systems.
- Specifying a custom membership provider, which allows you to substitute your own code to manage membership and maintain membership data in a custom data store

Membership, Roles, and the User Profile Properties

Although membership is a self-standing feature in ASP.NET for authentication, it can be integrated with ASP.NET role management to provide authorization services for your site. Membership can also be integrated with user profile properties to provide application-specific customization that can be tailored to individual users. For details, see Managing Authorization Using Roles and ASP.NET Profile Properties Overview. For information about how to use membership with roles, see Walkthrough: Managing Web Site Users with Roles.

How Membership Works

To use membership, you must first configure it for your site. The following are the basic steps you follow in order to configure membership:

1. Specify membership options as part of your Web site configuration. By default, membership is enabled. You can also specify what membership provider you want to use. The default provider

stores membership information in a Microsoft SQL Server database. However, you can use other providers as well, such as a provider for Windows Live ID. You can also choose to use Active Directory to store membership information, or you can specify a custom provider. For information about membership configuration options that can be specified in the Web.config file for your ASP.NET application, see Configuring an ASP.NET Application to Use Membership.

2. Configure your application to use forms authentication (as distinct from Windows or Windows Live ID authentication).

   You can also specify that some pages or folders in your application are protected and are accessible only to authenticated users. For more information, see Walkthrough: Managing Web Site Users with Roles.

3. Define user accounts for membership. You can do this in a variety of ways. You can use the Web Site Administration Tool, which provides a wizard-like interface for creating new users. Alternatively, you can use an ASP.NET Web page where you collect a user name and password (and optionally an email address), and then use the CreateUser membership method to create a new user in the membership system.

You can then use membership to authenticate users in your application. Typically, you will provide a page where users can log in, a registration page where new users can sign up, and a change-password page where users can change their password. The ASP.NET login controls (Login, LoginView, LoginStatus, LoginName, and PasswordRecovery) encapsulate virtually all of the logic required to prompt users for credentials and validate the credentials in the membership system.

The default template in Visual Studio for creating an ASP.NET project includes pages include much of this basic login functionality. For information about how to use the project templates, see Walkthrough: Creating an ASP.NET Web Site with Basic User Login. Alternatively, you can use the same ASP.NET controls that are in the template to create custom login and membership pages. For information about how to manually put together these ASP.NET controls to create an application that authenticates users, see Walkthrough: Creating a Web Site with Membership and User Login.

If you have configured the application to use forms authentication, ASP.NET will automatically display the login page if an unauthenticated user requests a protected page.

If you use the project templates that are provided in ASP.NET or use the login controls, they will automatically use the membership system to validate a user. If you create custom login controls, you must call the ValidateUser method to validate the user's user name and password. After the user is validated, information about the user can be persisted (for example, with an encrypted cookie if the user's browser accepts cookies). The login controls perform this task automatically, but if you create custom login controls, you can call methods of the FormsAuthentication class to create the cookie and write it to the user's computer. If a user has forgotten his or her password, the login page can call membership functions that help the user recover the password or reset the password.

Each time the user requests a page, ASP.NET forms authentication checks whether the user is authenticated. If a page is restricted, users who are authenticated and are members of the approved roles are allowed to view the page. Anonymous users (users who are not logged in) are directed to the login page. By default, the authentication cookie remains valid for the user's session.

After a user has been authenticated, the membership system makes available an object that contains information about the current user. For example, you can get properties of the membership user object to determine the user's name and email address, when the user last logged into your Web site, and so on.

An important aspect of the membership system is that you never need to explicitly perform any low-level database functions to get or set user information. For example, you create a new user by calling the

membership CreateUser method. The membership system handles the details of creating the necessary database records to store the user information. When you call the ValidateUser method to check a user's credentials, the membership system does all the database lookup for you.

See Also

Tasks
Walkthrough: Creating an ASP.NET Web Site with Basic User Login
Walkthrough: Creating a Web Site with Membership and User Login
Concepts
Securing Membership
Other Resources
Managing Users by Using Membership

Community Additions

That was not a support question

Thomas,

That was not a support question - it was a warning to other developer's of a potential issue that may have a serious impact on their implementations.

I for one appreciate such warnings - it at least lets me know I should check migrations carefully before doing so in production environments (of course that should be done anyway).


Rob Grainger
8/8/2013

password maze

hi

live 1234 $

yahoo 12345$ (domain name + string of digits (this same for all domains, think change)+ spl chr
bye


simpletense
5/5/2013

upgrade to .net 4 is buggy

Don't plan to upgrade your web app from .net2 to .net4 if you are using asp.net membershipprovider.

I lost all my user's passwords in a production environment by doing so. The trick to re-apply default SHA1 algorythm in the web.config on machineKey and membership didn't work at all..

Now I must tell 200 people we have lost their password, and no way to explain them this is your fault. You guys reinvent the wheel every 2 years, Just crazy and irresponsible !!

[tfl - 02 04 12] Hi - and thanks for your post. Community content is not the appropriate place for technical support queries. Instead, you should visit the MSDN Forums at http://forums.microsoft.com/MSDN, where such posts are welcomed and where you stand a much better chance of getting your query resolved. Sorry if that's not the answer you wanted to hear.

Thomas Lee
4/2/2012