Managing Users by Using Membership

ASP.NET membership enables you to validate and manage user information for your Web application. It provides functionality for validating user credentials, creating and modifying membership users, and managing user settings such as passwords and e-mail addresses. ASP.NET membership is primarily intended for use with ASP.NET forms authentication, but can be used anywhere within an ASP.NET application.

ASP.NET membership enables you to manage user authentication for your application while keeping the user information in the data source of your choice. Because ASP.NET membership uses providers to the membership data source, it does not require extensive code to read and write membership information.

ASP.NET membership consists primarily of built-in membership providers, which communicate with the data source, and the static [Membership](https://msdn.microsoft.com/en-us/library/system.web.security.membership(v=vs.100).aspx) class that exposes the functionality of the membership providers. You call the [Membership](https://msdn.microsoft.com/en-us/library/system.web.security.membership(v=vs.100).aspx) class from your ASP.NET code to perform user validation and management.

Introduction to Membership

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 or 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

Each time the user requests another protected page, ASP.NET Forms authentication checks whether the user is authenticated and then either allows the user to view the page or redirects the user 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 e-mail address, when the user last logged into your application, and so on.

## [Membership Configuration and Management](javascript:void(0))

You configure the membership system in your application's Web.config file. The easiest way to configure and manage membership is with the Web Site Administration Tool, which provides a wizard-based interface. As part of membership configuration, you specify:

* What membership provider to use. (This typically also specifies what database to store membership information in.)
* Password options such as encryption and whether to support password recovery based on a user-specific question.
* Users and passwords. If you are using the Web Site Administration Tool, you can create and manage users directly. Otherwise, you must call membership functions to create and manage users programmatically.

Configuring an ASP.NET Application to Use Membership

ASP.NET Membership is configured using the [membership](https://msdn.microsoft.com/en-us/library/1b9hw62f(v=vs.100).aspx) element in the Web.config file for your application. The membership element is a sub-element of the system.web section. You can enable ASP.NET Membership for an application by directly editing the Web.config file for that application, or you can use the Web Site Administration Tool, which provides a wizard-based interface. As part of membership configuration,

For an example of configuring ASP.NET Membership using the Web Site Administration Tool, see [Walkthrough: Creating a Web Site with Membership and User Login](https://msdn.microsoft.com/en-us/library/879kf95c(v=vs.100).aspx).

[Specifying the Default Provider](javascript:void(0))

You specify the default membership provider using the defaultProvider attribute of the membership element. The machine configuration specifies a [SqlMembershipProvider](https://msdn.microsoft.com/en-us/library/system.web.security.sqlmembershipprovider(v=vs.100).aspx) instance named "AspNetSqlMembershipProvider" that is identified as the default provider if you do not specify a default provider explicitly. The "AspNetSqlMembershipProvider" connects to the aspnetdb database in the local SQL Server.