

Contents

- [Problem](#)
- [Solution](#)
 - [Forms authentication](#)
 - [Windows authentication](#)
 - [Troubleshooting](#)
- [Prerequisites](#)

This sample application illustrates authentication using **ASP.NET membership and roles**.

- **Platform:** Silverlight
- **Language:** C#, VB
- **Download:** [ASP.NET membership and roles \(Silverlight\)](#)

Problem

You'd like a small sample Silverlight application which illustrates the use of ASP.NET security features.

Solution

The sample here is provided if all you need is a simple example to see the basics of using ASP.NET security in your application. The sample shows both Forms and Windows authentication options, and can help get you started in understanding the basics of using ASP.NET security in Silverlight.

Forms authentication

To try Forms authentication, be sure that the **<authentication>** element in system.web reads **Forms**.

Run the application (be sure that the web project is the startup project).

If you choose the **Manager Login** option and click **Login**, you will be logged in as the "Manager" user. The manager here is also a member of the "Admin" role, which allows for both fetching and saving privileges in this sample. See the Login method in Page.xaml.cs for more information.

Some simple authorization checks are provided in the *EntityServerEventsHandler* class in the web project.

In the *EntityServerQueryInterceptor* you see a programmatic check and modification of the query using a Filter based on the user's role. In the *EntityServerSaveInterceptor* you see a programmatic check requiring that the user have admin privileges in order to save data.

If you login with the **Employee Login** option you'll see that the Save button has been disabled. See the LoginCompleted callback handler in Page.xaml.cs for the simple enabling/disabling logic it performs based on the current user information. Note that disabling UI features will not secure your application, it just makes for a more intuitive UI. All authorization checks should be performed on the server.

Windows authentication

To try Windows authentication, be sure that the **<authentication>** element in system.web says "Windows".

Before pressing the Login button, choose the "Windows Authentication" login option. You should be authenticated with your Windows account and have fetch privileges. You'll note that a LoginAsync call is still required, although no credentials are passed. The Login call is how DevForce obtains the security token it uses for all other requests to the server.

Troubleshooting

- If the membership database is not found or does not contain the user's information, the *IsAuthenticated* value displayed on the page is always *False*. If you perform a Windows login while using Forms authentication this will also occur, since your Windows account is not in the database.
- When trying to login, the **Error using ASP.NET Membership** message will display if SQL Express is not installed or the service has not been started. The remainder of the message will indicate the exact cause.

Prerequisites

This sample is set up to access a local ASP.NET membership database named aspnetdb on an instance of SQL Express. This database contains membership, role, and profile information for two users: "Manager" and "Employee", and is located in a file, aspnetdb.mdf, stored in the Data folder (adjacent to the CodeCS folder).

If you wish to use SQL Server rather than SQL Express you can do so by attaching the aspnetdb.mdf database to SQL Server and changing the *LocalSqlServer* connection string in the web.config. There are comments in web.config (near the bottom) to show you the necessary change.

The aspnetdb.mdf file used in this sample is copied to the App_Data folder upon first build (see the pre-build event on the web project). It expects to find the file in the Data folder adjacent to CodeCS. If you move the solution, be sure to move the Data folder as well.

You can use the **ASP.NET Configuration** menu item from the **Project** menu to view and edit membership and role information in the database.