



# ASP.NET Core Identity

Hans-Petter Halvorsen

# ASP.NET Core

If you have never used ASP.NET Core, I suggest the following Videos:

- ASP.NET Core - Hello World  
<https://youtu.be/lcQsWYgQXK4>
- ASP.NET Core – Introduction  
<https://youtu.be/zkOtiBcwo8s>

**ASP.NET Core Resources** (Textbook, Videos, Code, etc.):  
<https://halvorsen.blog/documents/programming/web/aspnet>

# ASP.NET Core Identity

- An API that supports user interface (UI) login functionality out of the box.
- You can manage users, passwords, roles, email confirmation, 2FA, and more.
- Users can create an account with the login information stored in Identity or they can use an external login provider.
- Supported external login providers include Facebook, Google, Microsoft Account, and Twitter.

# ASP.NET Core Identity

- Introduction to Identity on ASP.NET Core:  
<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity>
- Scaffold Identity in ASP.NET Core projects:  
<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/scaffold-identity>

# Create Project in Visual Studio with Identity Enabled

# Visual Studio

## Create a new project

### Recent project templates

- ASP.NET Core Web Application C#
- Class Library (.NET Core) C#
- Blank Solution
- Windows Forms App (.NET Core) C#
- Windows Forms App (.NET Framework) C#
- NI Windows Forms Application
- ASP.NET Web Application (.NET Framework) C#
- ASP.NET Web Application (.NET Framework) Visual Basic
- Python Application Python

Search for templates (Alt+S)

[Clear all](#)

C# Windows All project types



#### Console App (.NET Core)

A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.

C# Linux macOS Windows Console



#### ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C# Linux macOS Windows Cloud Service Web



#### Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly. These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C# Linux macOS Windows Cloud Web



#### Class Library (.NET Standard)

A project for creating a class library that targets .NET Standard.

C# Android iOS Linux macOS Windows Library



#### gRPC Service

A project template for creating a gRPC ASP.NET Core service using .NET Core.

C# Linux macOS Windows Cloud Service Web



#### Razor Class Library

Back

Next

# Select Authentication

Create a new ASP.NET Core web ap...

.NET Core

ASP.NET Core 3.1

Empty

An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

API

A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

Web Application

A project template for creating an ASP.NET Core application with example ASP.NET Razor Pages content.

Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support

(Requires [Docker Desktop](#))

Linux

Author: Microsoft

Source: .NET Core 3.1.1

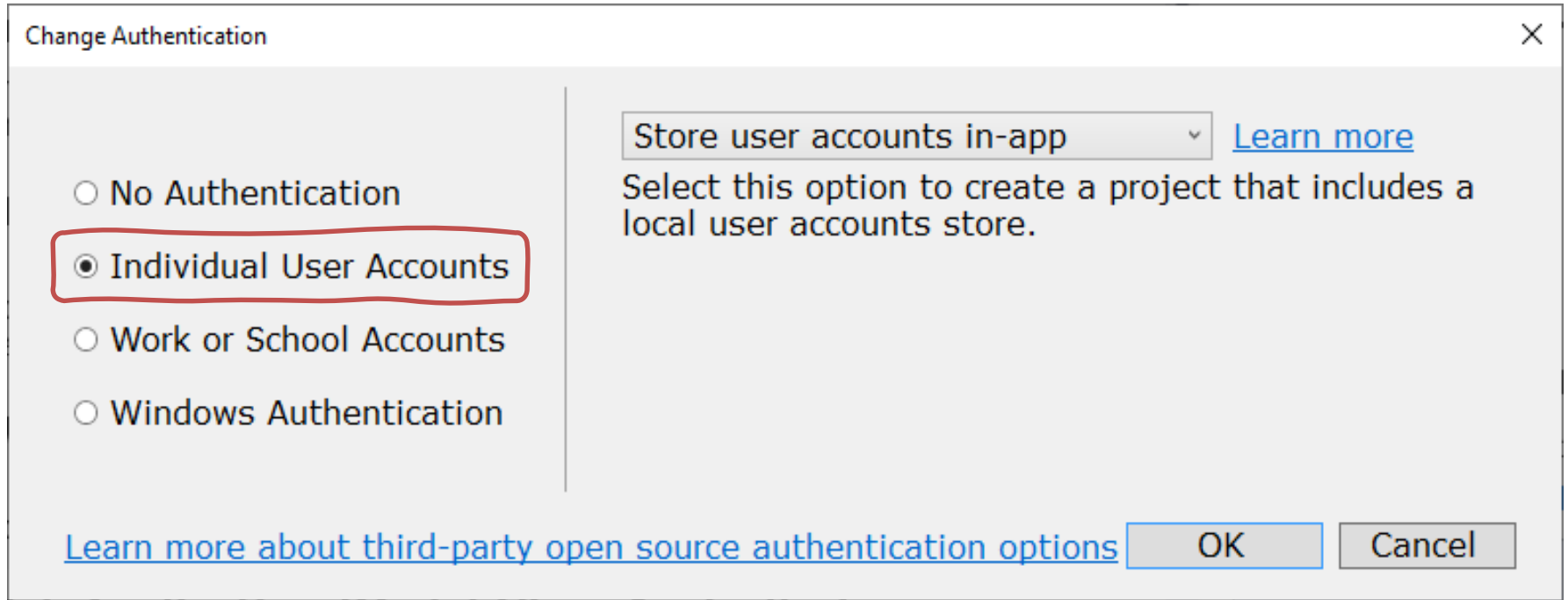
[Get additional project templates](#)

Back

Create

7

# Change Authentication



A screenshot of a 'Change Authentication' dialog box. The dialog has a title bar with the text 'Change Authentication' and a close button (X) in the top right corner. On the left side, there are four radio button options: 'No Authentication', 'Individual User Accounts' (which is selected and highlighted with a red rounded rectangle), 'Work or School Accounts', and 'Windows Authentication'. On the right side, there is a dropdown menu showing 'Store user accounts in-app' with a downward arrow, followed by a blue link 'Learn more'. Below the dropdown, there is a text description: 'Select this option to create a project that includes a local user accounts store.' At the bottom left, there is a blue link 'Learn more about third-party open source authentication options'. At the bottom right, there are two buttons: 'OK' and 'Cancel'.

Change Authentication

☐ No Authentication

☒ Individual User Accounts

☐ Work or School Accounts

☐ Windows Authentication

Store user accounts in-app [Learn more](#)

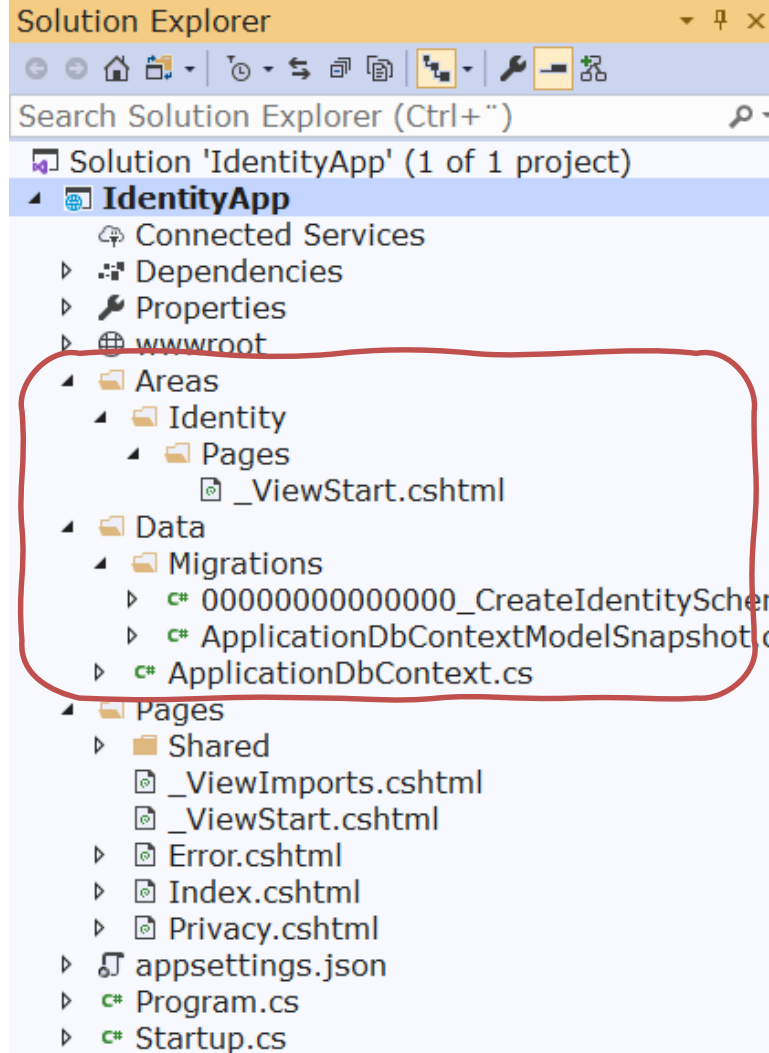
Select this option to create a project that includes a local user accounts store.

[Learn more about third-party open source authentication options](#)



# Solution Explorer

Lots of new stuff has been created for us that are used by the Identity features we have enabled



# Create Identity Database

# You need to create the Identity Database

A database operation failed while processing the request.

SqlException: Cannot open database "aspnet-IdentityApp-53bc9b9d-9d6a-45d4-8429-2a2761773502" requested by the login. The login failed. Login failed for user 'XPS15HPH\hansha'.

Applying existing migrations for ApplicationDbContext may resolve this issue

There are migrations for ApplicationDbContext that have not been applied to the database

- 0000000000000000\_CreatelIdentitySchema

## Apply Migrations

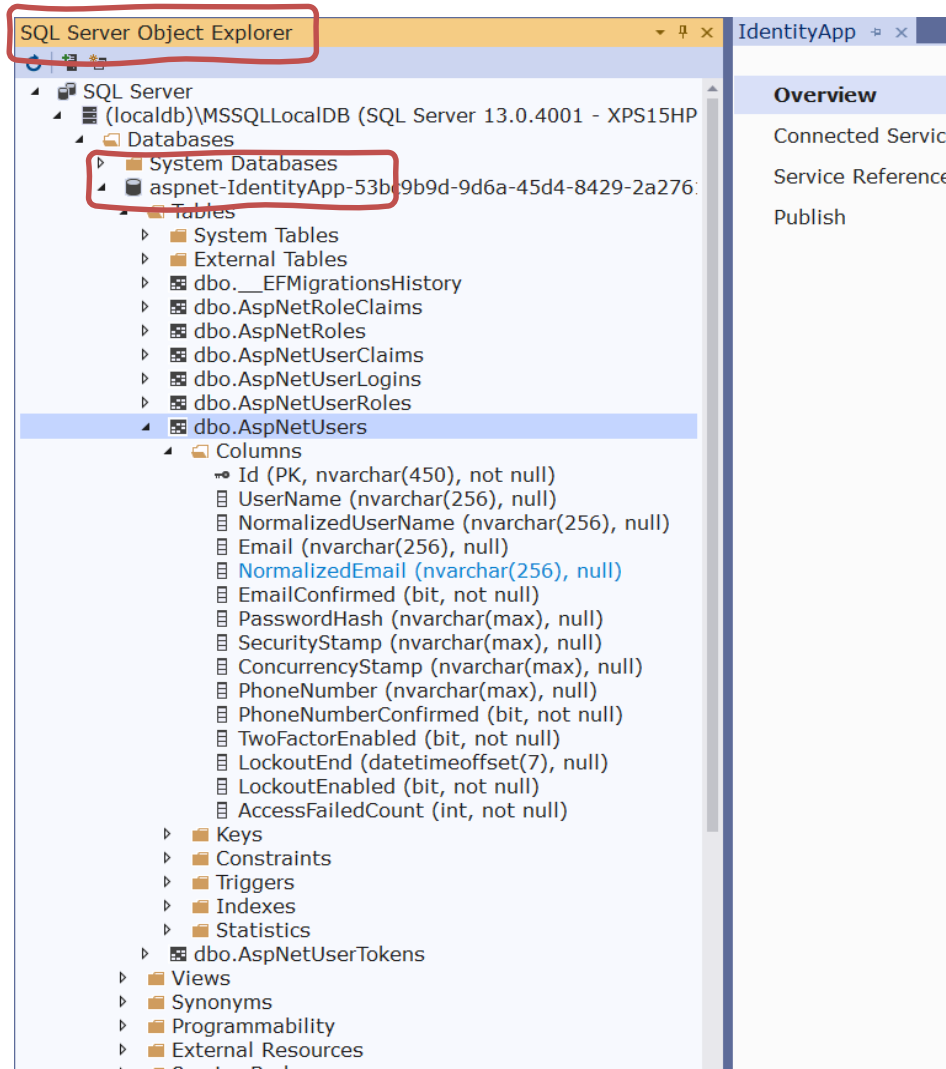
In Visual Studio, you can use the Package Manager Console to apply pending migrations to the database:

```
PM> Update-Database
```

Alternatively, you can apply pending migrations from a command prompt at your project directory:

```
> dotnet ef database update
```

# Identity Database



The screenshot displays the SQL Server Object Explorer interface. The 'SQL Server Object Explorer' window is open, showing the hierarchy of the local database. The 'aspnet-IdentityApp-53bc9b9d-9d6a-45d4-8429-2a276' database is selected, and its 'dbo.AspNetUsers' table is expanded. The table's columns are listed, including 'Id' (PK), 'UserName', 'NormalizedUserName', 'Email', 'NormalizedEmail', 'EmailConfirmed', 'PasswordHash', 'SecurityStamp', 'ConcurrencyStamp', 'PhoneNumber', 'PhoneNumberConfirmed', 'TwoFactorEnabled', 'LockoutEnd', 'LockoutEnabled', and 'AccessFailedCount'. The 'Overview' tab on the right shows the 'Connected Services' and 'Service References' sections.

SQL Server Object Explorer

SQL Server

(localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - XPS15HP)

Databases

System Databases

aspnet-IdentityApp-53bc9b9d-9d6a-45d4-8429-2a276

Tables

System Tables

External Tables

dbo.\_\_EFMigrationsHistory

dbo.AspNetRoleClaims

dbo.AspNetRoles

dbo.AspNetUserClaims

dbo.AspNetUserLogins

dbo.AspNetUserRoles

dbo.AspNetUsers

Columns

Id (PK, nvarchar(450), not null)

UserName (nvarchar(256), null)

NormalizedUserName (nvarchar(256), null)

Email (nvarchar(256), null)

NormalizedEmail (nvarchar(256), null)

EmailConfirmed (bit, not null)

PasswordHash (nvarchar(max), null)

SecurityStamp (nvarchar(max), null)

ConcurrencyStamp (nvarchar(max), null)

PhoneNumber (nvarchar(max), null)

PhoneNumberConfirmed (bit, not null)

TwoFactorEnabled (bit, not null)

LockoutEnd (datetimeoffset(7), null)

LockoutEnabled (bit, not null)

AccessFailedCount (int, not null)

Keys

Constraints

Triggers

Indexes

Statistics

dbo.AspNetUserTokens

Views

Synonyms

Programmability

External Resources

Overview

Connected Services

Service References

Publish

# Register New Account and Log In

# You are now ready

- Create Account
- Log In
- Etc.

# Register – Create New Account

IdentityApp [Home](#) [Privacy](#)

[Register](#) [Login](#)

## Register

Create a new account.

Email

Password

Confirm password

Register

Use another service to register.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

# Confirm Registration

IdentityApp Home Privacy

Register Login

## Register confirmation

This app does not currently have a real email sender registered, see [these docs](#) for how to configure a real email sender. Normally this would be emailed: [Click here to confirm your account](#)



# Log In

IdentityApp [Home](#) [Privacy](#)

[Register](#) [Login](#)

## Log in

Use a local account to log in.

Email

Password

☐ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

# You are Logged In

IdentityApp

[Home](#) [Privacy](#)

Hello [hans.p.halvorsen@usn.no!](#)

[Logout](#)

## Welcome

Learn about [building Web apps with ASP.NET Core](#).

# Manage your Account

IdentityApp [Home](#) [Privacy](#)

Hello hans.p.halvorsen@usn.no! [Logout](#)

## Manage your account

Change your account settings

Profile

Email

Password

Two-factor  
authentication

Personal data

### Profile

Thank you for confirming your email. ×

Username

hans.p.halvorsen@usn.no

Phone number

Save

# 2FA

## 2 Factor Authentication

# 2 Factor Authentication

- All modern systems offer what we call “2 Factor Authentication”
- This means in addition to enter the password, the user needs to enter a one-time password received on E-Mail or SMS.
- An alternative is to use an Authenticator App like “Google Authenticator” or “Microsoft Authenticator” available on iPhone and Android.

# Start using 2FA

IdentityApp Home Privacy

Hello hans.p.halvorsen@usn.no! Logout

## Manage your account

### Change your account settings

[Profile](#)

[Email](#)

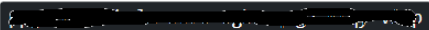
[Password](#)

**Two-factor authentication**

[Personal data](#)

### Configure authenticator app

To use an authenticator app go through the following steps:

1. Download a two-factor authenticator app like Microsoft Authenticator for [Android](#) and [iOS](#) or Google Authenticator for [Android](#) and [iOS](#).
2. Scan the QR Code or enter this key  into your two factor authenticator app. Spaces and casing do not matter.

[Learn how to enable QR code generation.](#)

3. Once you have scanned the QR code or input the key above, your two factor authentication app will provide you with a unique code. Enter the code in the confirmation box below.

Verification Code

Verify

# Log In using 2FA

IdentityApp [Home](#) [Privacy](#)

[Register](#) [Login](#)

## Two-factor authentication

Your login is protected with an authenticator app. Enter your authenticator code below.

Authenticator code

☐ Remember this machine

Log in

Enter Code from  
Authentication App

Don't have access to your authenticator device? You can [log in with a recovery code](#).

# Application



# Start Creating your Application

- When the Identity features are installed, configured and set up, you can start creating the rest of your application
- Typically you need to check in your different web pages if the user is logged in (authenticated) or not

# @User.Identity

Typically you want to use the following:

- @User.Identity.IsAuthenticated
- @User.Identity.Name

# Check if you are Logged In or not in your Code

IdentityApp [Home](#) [Privacy](#) [Register](#) [Login](#)

---

## Welcome

ASP.NET Core Identity Example Application

Is Authenticated: False  
UserName:

You are NOT Logged In

# Check if you are Logged In or not in your Code

IdentityApp [Home](#) [Privacy](#)

Hello hans.p.halvorsen@usn.no! [Logout](#)

## Welcome

ASP.NET Core Identity Example Application

Is Authenticated: True

UserName: hans.p.halvorsen@usn.no

You are Logged In

# Razor Code

```
@page
@model IndexModel
@{
    ViewData["Title"] = "Home page";
}
```

```
<div>
    <h1>Welcome</h1>

    <p>ASP.NET Core Identity Example Application</p>
```

```
    <p>Is Authenticated: @User.Identity.IsAuthenticated</p>
```

```
    <p>UserName: @User.Identity.Name</p>
```

```
</div>
```

# Welcome

ASP.NET Core Identity Example Application

Is Authenticated: False

UserName:

## Data

You are not Logged In

# Welcome

ASP.NET Core Identity Example Application

Is Authenticated: True

UserName: hans.p.halvorsen@usn.no

## Data

Show Data here (e.g., from your Database)...

...

<div>

# Razor Code

<h2>Data</h2>

**@if (User.Identity.IsAuthenticated)**

{

<p>Show Data here (e.g., from your Database)...</p>

}

else

{

<p>You are not Logged In</p>

}

</div>

# Scaffolding



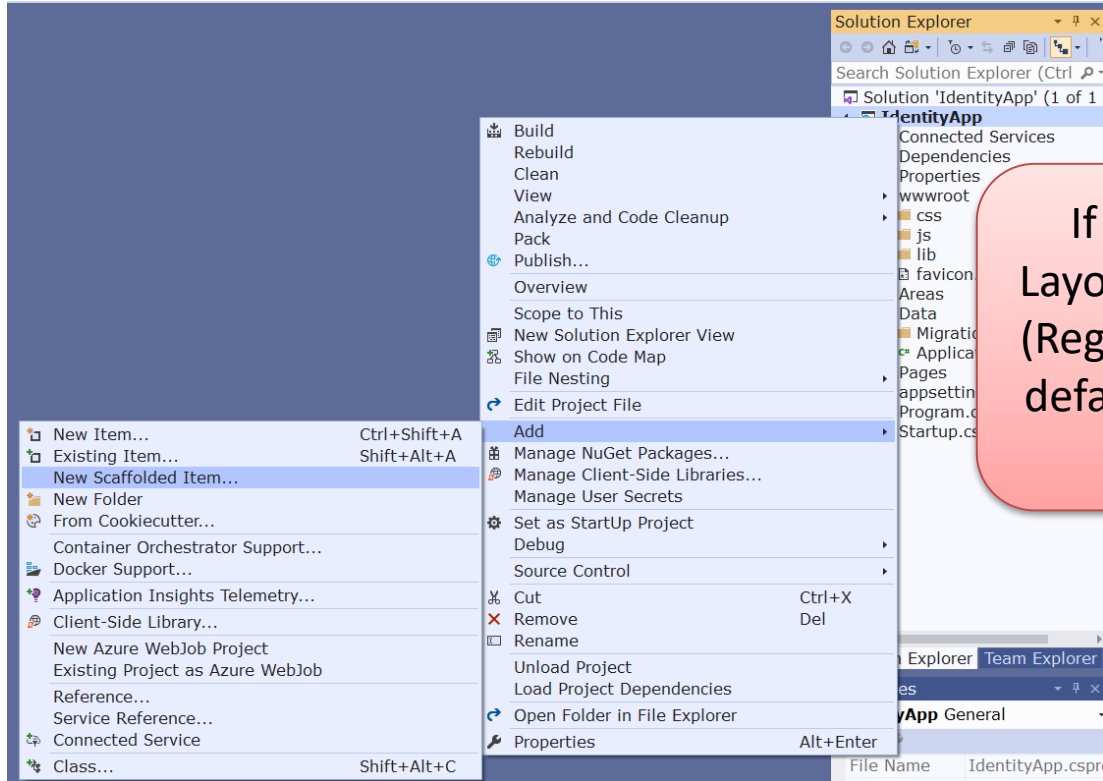
# Scaffolding

- Scaffolding, also called scaffold or staging is a temporary structure used to support a work crew and materials to aid in the construction, maintenance and repair of buildings, etc.
- <https://en.wikipedia.org/wiki/Scaffolding>
- Scaffolding, as used in computing, refers to one of two techniques: The first is a code generation technique related to database access in some model–view–controller frameworks; the second is a project generation technique supported by various tools.
- [https://en.wikipedia.org/wiki/Scaffold\\_\(programming\)](https://en.wikipedia.org/wiki/Scaffold_(programming))

# Scaffold Identity in ASP.NET Core Projects

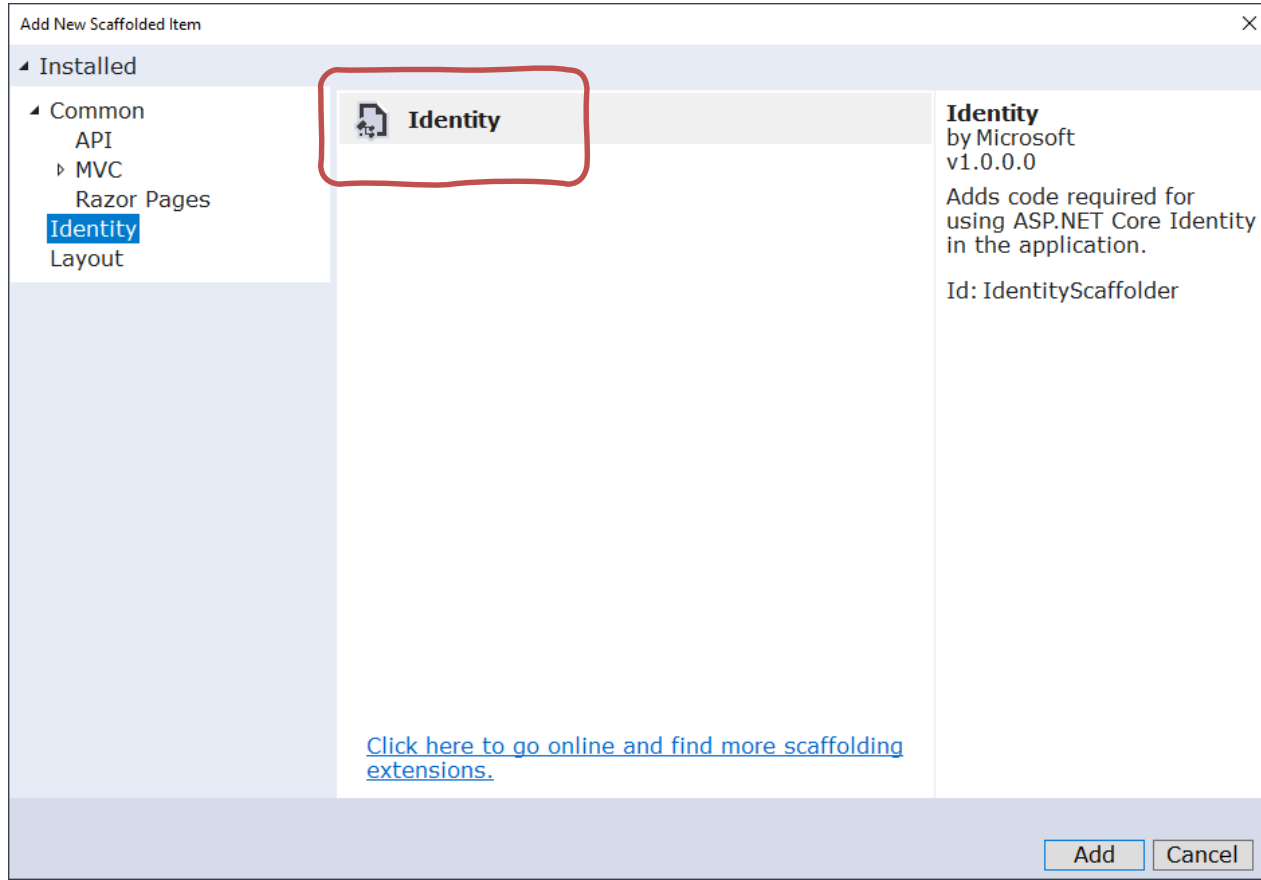
- Applications that include Identity can apply the scaffolder to selectively add the source code contained in the Identity Razor Class Library (RCL).
- **You might want to generate source code so you can modify the code and change the behavior.**
- **For example, you could instruct the scaffolder to generate the code used in login or registration.**
- Generated code takes precedence over the same code in the Identity RCL.

# Scaffold Identity in ASP.NET Core Projects



If you are not happy with the default Layout of the different Identity web pages (Register, Login, etc.) You can override the default Scaffolding and modify these web pages, so they fit your needs

# Add Identity Scaffolded Item



# Scaffold Identity in ASP.NET Core Projects

Add Identity ×

Select an existing layout page, or specify a new one:

~/Pages/Shared/\_Layout.cshtml

Select the Identity page(s) you want to override

(Leave empty if it is set in a Razor \_viewstart file)

☐ Override all files

Choose files to override

<input type="checkbox"/> Account\StatusMessage	<input type="checkbox"/> Account\AccessDenied	<input type="checkbox"/> Account\ConfirmEmail
<input type="checkbox"/> Account\ConfirmEmailChange	<input type="checkbox"/> Account\ExternalLogin	<input type="checkbox"/> Account\ForgotPassword
<input type="checkbox"/> Account\ForgotPasswordConfirmation	<input type="checkbox"/> Account\Lockout	<input checked="" type="checkbox"/> Account>Login
<input type="checkbox"/> Account>LoginWith2fa	<input type="checkbox"/> Account>LoginWithRecoveryCode	<input type="checkbox"/> Account\Logout
<input type="checkbox"/> Account\Manage\Layout	<input type="checkbox"/> Account\Manage\ManageNav	<input type="checkbox"/> Account\Manage\StatusMessage
<input type="checkbox"/> Account\Manage\ChangePassword	<input type="checkbox"/> Account\Manage\DeletePersonalData	<input type="checkbox"/> Account\Manage\Disable2fa
<input type="checkbox"/> Account\Manage\DownloadPersonalData	<input type="checkbox"/> Account\Manage\Email	<input type="checkbox"/> Account\Manage\EnableAuthenticator
<input type="checkbox"/> Account\Manage\ExternalLogins	<input type="checkbox"/> Account\Manage\GenerateRecoveryCode	<input type="checkbox"/> Account\Manage\Index
<input type="checkbox"/> Account\Manage\PersonalData	<input type="checkbox"/> Account\Manage\ResetAuthenticator	<input type="checkbox"/> Account\Manage\SetPassword
<input type="checkbox"/> Account\Manage\ShowRecoveryCodes	<input type="checkbox"/> Account\Manage\TwoFactorAuthenticatio	<input type="checkbox"/> Account\Register
<input type="checkbox"/> Account\RegisterConfirmation	<input type="checkbox"/> Account\ResetPassword	<input type="checkbox"/> Account\ResetPasswordConfirmation

Data context class: IdentityApp.Data.IdentityAppContext +

☐ Use SQLite instead of SQL Server

User class: +

Add Cancel

FileEditViewProjectBuildDebugTestAnalyzeToolsExtensionsWindowHelpSearch (Ctrl+Q)

DebuAny CPUIIS Express

Live Share

ScaffoldingReadMe.txt

1Support for ASP.NET Core Identity was added to your project

2The code for adding Identity to your project was generated under Areas/Identity.

3

4Configuration of the Identity related services can be found in the Areas/Identity/IdentityHostingStartup.cs file.

5

6If your app was previously configured to use Identity, then you should remove the call to the AddIdentity method from your ConfigureServices method.

7

8The generated UI requires support for static files. To add static files to your app:

91. Call app.UseStaticFiles() from your Configure method

10

11To use ASP.NET Core Identity you also need to enable authentication. To authentication to your app:

121. Call app.UseAuthentication() from your Configure method (after static files)

13

14The generated UI requires MVC. To add MVC to your app:

151. Call services.AddMvc() from your ConfigureServices method

162. Call app.UseRouting() at the top your Configure method, and UseEndpoints() after authentication:

17app.UseEndpoints(endpoints =>

18{

19endpoints.MapControllers();

20endpoints.MapRazorPages();

21});

22

23The generated database code requires Entity Framework Core Migrations. Run the following commands:

241. dotnet ef migrations add CreateIdentitySchema

252. dotnet ef database update

26Or from the Visual Studio Package Manager Console:

271. Add-Migration CreateIdentitySchema

282. Update-Database

29

30Apps that use ASP.NET Core Identity should also use HTTPS. To enable HTTPS see <https://go.microsoft.com/fwlink/?linkid=848054>.

31

32

100 %

No issues found

Ln: 1Ch: 1SPC CRLF

Output

Show output from: Code Generation Log

Finding the generator 'identity'...

Running the generator 'identity'...

Solution Explorer

Search Solution Explorer (Ctrl+)

Solution 'IdentityApp' (1 of 1 project)

IdentityApp

Connected Services

Dependencies

Properties

wwwroot

Areas

Identity

Data

Pages

Account

\_ViewImports.cshtml

Login.cshtml

\_ValidationScriptsPartial

\_ViewImports.cshtml

\_ViewStart.cshtml

IdentityHostingStartup.cs

Data

Migrations

ApplicationDbContext.cs

Pages

Shared

\_ViewImports.cshtml

\_ViewStart.cshtml

Error.cshtml

Index.cshtml

Privacy.cshtml

appsettings.json

Program.cs

ScaffoldingReadMe.txt

Startup.cs

Properties

IdentityApp General

Full PathC:\Users\hansp\On

Project FolderC:\Users\hansp\On

UserSecretsIdaspnet-IdentityAp

UserSecretsId

Ready

Add to Source Control

# Existing Login.cshtml

IdentityApp Home Privacy

Register Login

## Log in

Use a local account to log in.

Email

Password

☐ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

I want to remove this

# Updated Login.cshtml

IdentityApp [Home](#) [Privacy](#)

[Register](#) [Login](#)

## Log in

Use a local account to log in.

Email

Password

☐ Remember me?

Log in

[Forgot your password?](#)

[Register as a new user](#)





# Resources

- Introduction to Identity on ASP.NET Core:  
<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity>
- Scaffold Identity in ASP.NET Core projects:  
<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/scaffold-identity>
- Account confirmation and password recovery in ASP.NET Core:  
<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/accconfirm>

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

