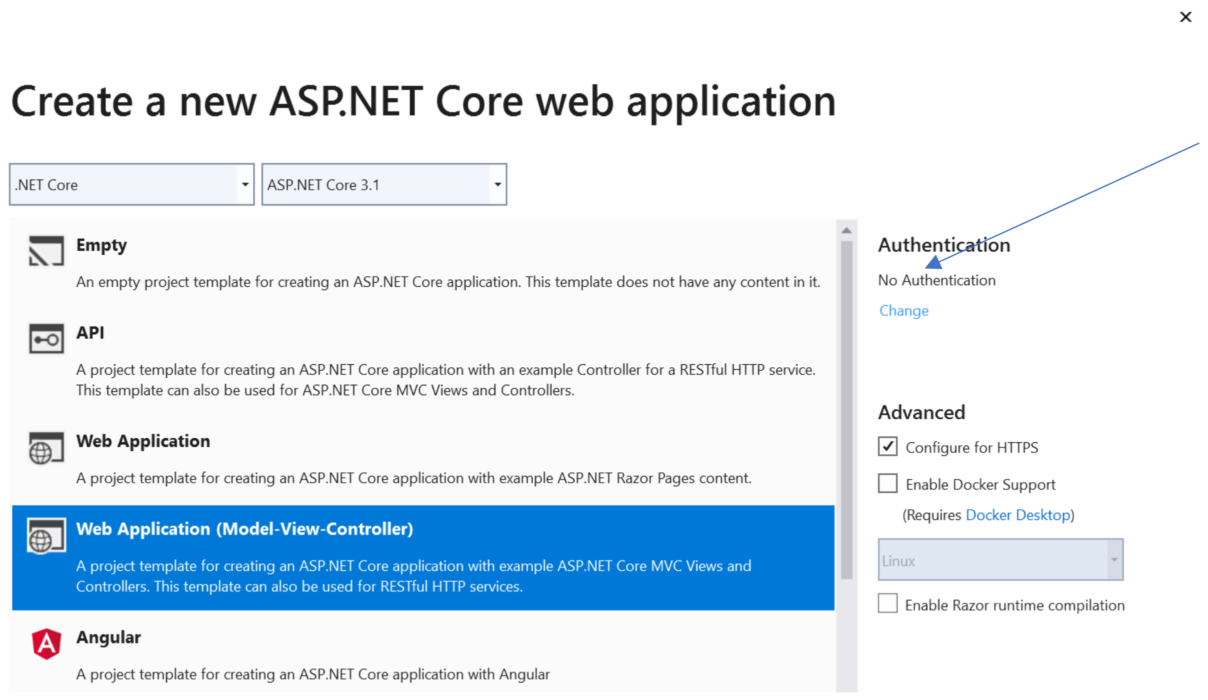
Create a new project with authentication



Create a new ASP.NET Core web application

.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.

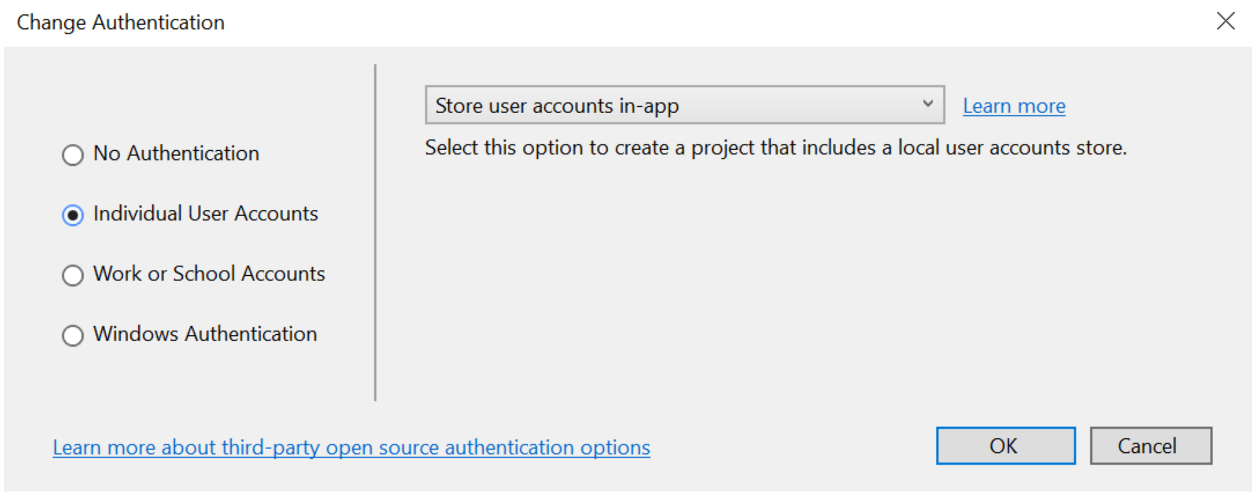
Web Application (Model-View-Controller)
A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

Angular
A project template for creating an ASP.NET Core application with Angular

Authentication
No Authentication
Change

Advanced
☑ Configure for HTTPS
☐ Enable Docker Support
(Requires Docker Desktop)
Linux
☐ Enable Razor runtime compilation

Change Authentication upon creating a new project



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

OK | Cancel

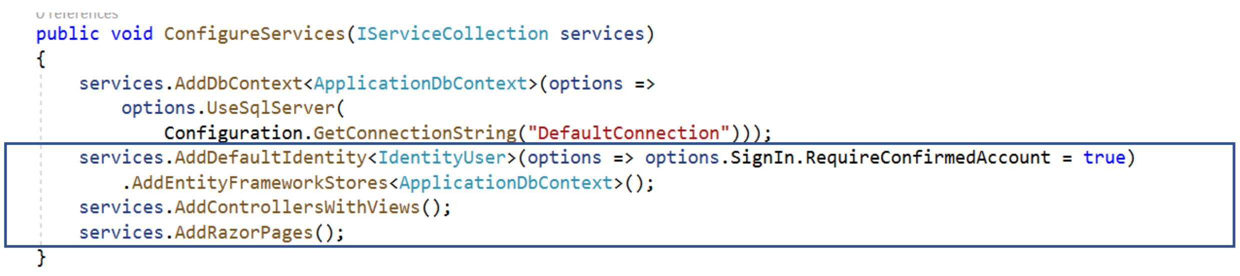
You can choose between

- **Store user accounts in-app**: includes a local user accounts store
- **Connect to an existing user store in the cloud**: connect to an existing Azure AD B2C application

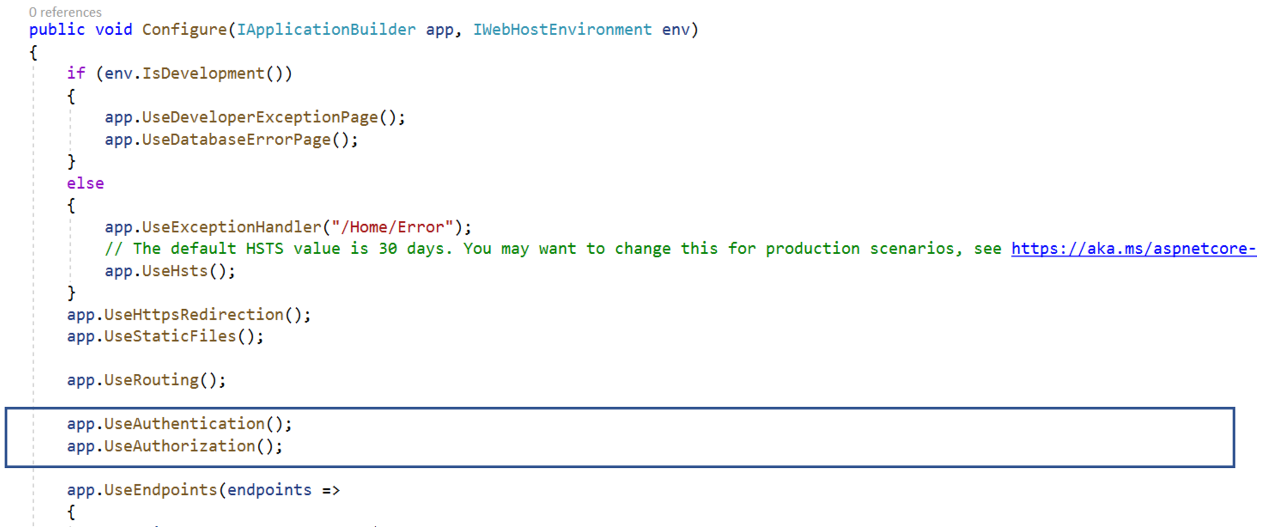
Note that the types of authentication are listed below.

- **None**: No authentication
- **Individual**: Individual authentication
- **IndividualB2C**: Individual authentication with Azure AD B2C
- **SingleOrg**: Organizational authentication for a single tenant
- **MultiOrg**: Organizational authentication for multiple tenants
- **Windows**: Windows authentication

Adding authentication will modify the Startup.cs file

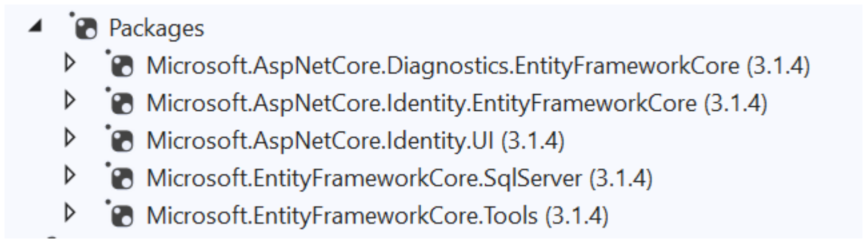


```
0 references
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));
    services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
        .AddEntityFrameworkStores<ApplicationDbContext>();
    services.AddControllersWithViews();
    services.AddRazorPages();
}
```



```
0 references
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
        app.UseDatabaseErrorPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-
        app.UseHsts();
    }
    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthentication();
    app.UseAuthorization();

    app.UseEndpoints(endpoints =>
    {
```

- **Verify the installed packages**

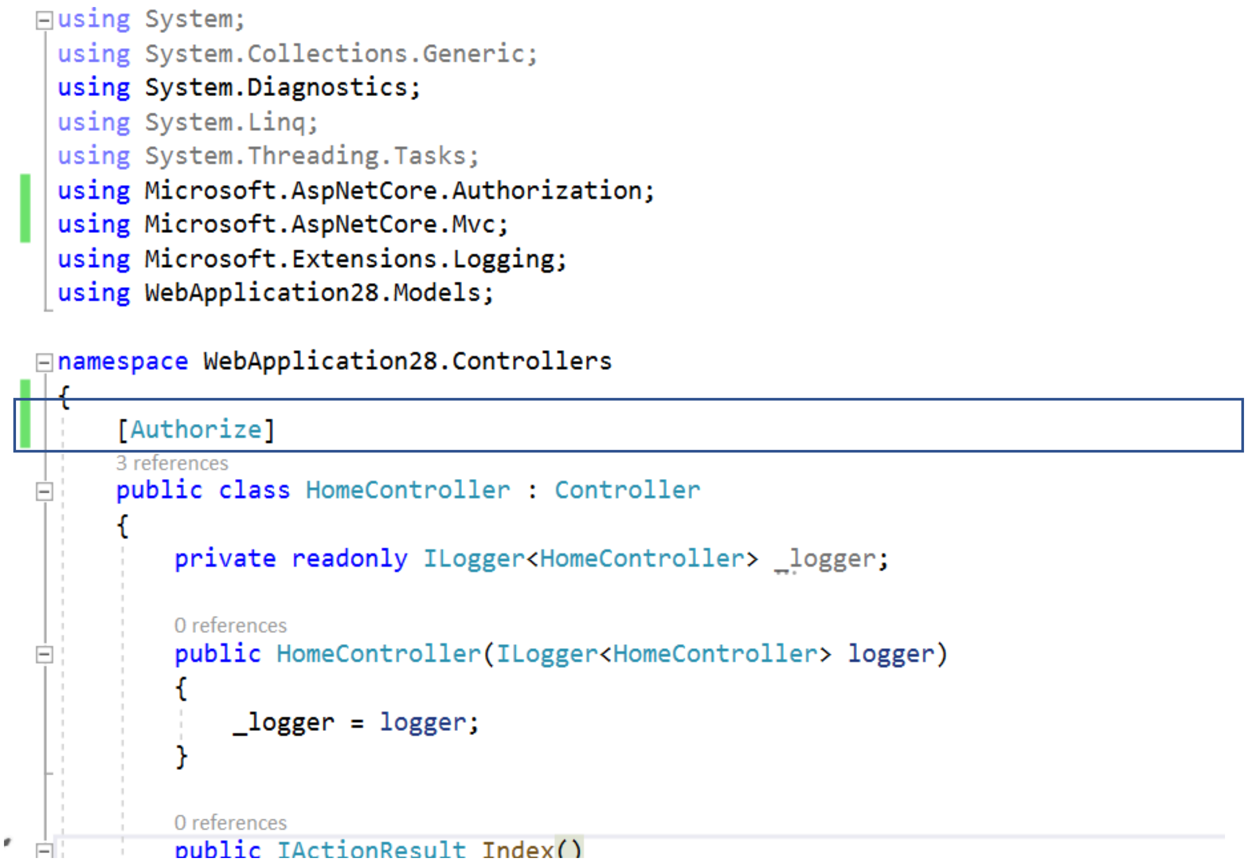

Packages
Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore (3.1.4)
Microsoft.AspNetCore.Identity.EntityFrameworkCore (3.1.4)
Microsoft.AspNetCore.Identity.UI (3.1.4)
Microsoft.EntityFrameworkCore.SqlServer (3.1.4)
Microsoft.EntityFrameworkCore.Tools (3.1.4)

- update also your **ConfigureServices()** method in your Startup.cs class, by calling **AddRazorPagesOptions()** after **AddMvc()**, this will helps to add Authorization for your pages , so ensures that the CRUD pages for Creating, Editing and Deleting any of the LearningResources are only accessible to someone who is currently logged in.

- Razor Pages have multiple ways of restricting access to pages and folders, including the following methods:

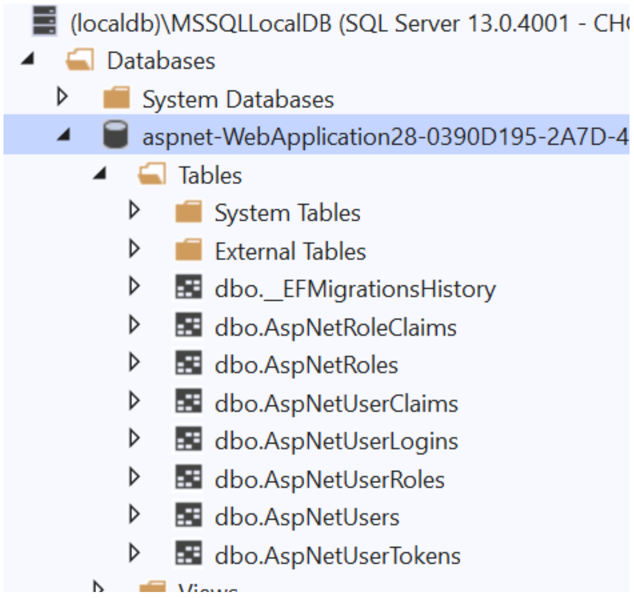
  - AuthorizePage: Require authorization to access a page
  - AuthorizeFolder: Require authorization to access a folder of pages
  - AuthorizeAreaPage: Require authorization to access an area page
  - AuthorizeAreaFolder: Require authorization to access a folder of areas
  - AllowAnonymousToPage: Allow anonymous access to a page
  - AllowAnonymousToFolder: Allow anonymous access to a folder of pages

- In order to restrict specific parts of the application, we will implement Authorization in our app.



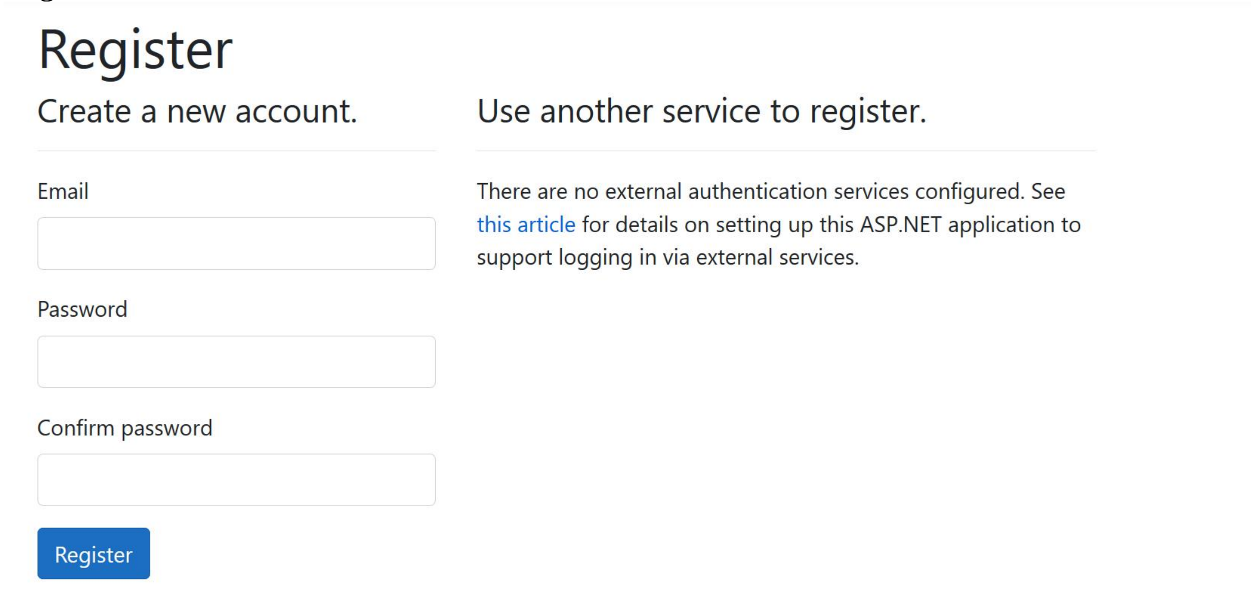
```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using WebApplication28.Models;

namespace WebApplication28.Controllers
{
    [Authorize]
    3 references
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        0 references
        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        0 references
        public IActionResult Index()
```

- Now update the database using the command : update-database.
- Check the created tables



```
(localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - CH
  Databases
    System Databases
    aspnet-WebApplication28-0390D195-2A7D-4
      Tables
        System Tables
        External Tables
        dbo.__EFMigrationsHistory
        dbo.AspNetRoleClaims
        dbo.AspNetRoles
        dbo.AspNetUserClaims
        dbo.AspNetUserLogins
        dbo.AspNetUserRoles
        dbo.AspNetUsers
        dbo.AspNetUserTokens
```

- Run your application to view it
- You can check if you can view any pages (privacy) if you are nor registered
- Register a new user



Register

Create a new account.

Email

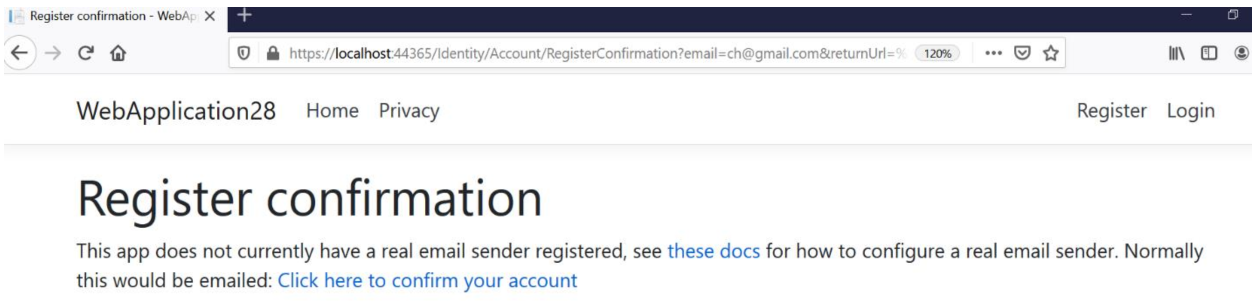
Password

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.



https://localhost:44365/Identity/Account/RegisterConfirmation?email=ch@gmail.com&returnUrl=%

WebApplication28 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

- Confirm your account, before you try to login