Individual Accounts

## ASPNETCoreWebAppRazorPagesMicrosoftIdentityPlatform



* appsettings.json add Connection String
  + public class ApplicationDbContext : IdentityDbContext {
* Data/Migrations with EF migrations
  + Tables for Users, Roles..
* Login page
* Program.cs
  + var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");
  + builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(connectionString));
  + builder.Services.AddDatabaseDeveloperPageExceptionFilter();
  + builder.Services
  + .AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
  + .AddEntityFrameworkStores<ApplicationDbContext>();
  + builder.Services.AddRazorPages();

Microsoft Identity Platform

## ASPNETCoreWebAppRazorPagesMicrosoftIdentityPlatform

A screenshot of a computer

Description automatically generated

* appsettings.json add Connection String
* Login page
* Program.cs
  + builder.Services

.AddAuthentication(OpenIdConnectDefaults.AuthenticationScheme)

.AddMicrosoftIdentityWebApp(builder.Configuration.GetSection("AzureAd"));

* + builder.Services.AddAuthorization(options => {

// By default, all incoming requests will be authorized according to the default policy.

options.FallbackPolicy = options.DefaultPolicy;

});

* + builder.Services.AddRazorPages().AddMicrosoftIdentityUI();

Windows

## ASPNETCoreWebAppRazorPagesWindows

A screenshot of a computer

Description automatically generated

* Program.cs
  + builder.Services

.AddAuthentication(NegotiateDefaults.AuthenticationScheme)

.AddNegotiate();

* + builder.Services.AddAuthorization(options => {

// By default, all incoming requests will be authorized according to the default policy.

options.FallbackPolicy = options.DefaultPolicy;

});

* + builder.Services.AddRazorPages();