CURSO ASP NET CORE - BLAZOR WEB, AUTENTICAÇÃO E AUTORIZAÇÃO EM APIs

Projeto inicial do curso: <https://github.com/alura-cursos/ScreenSound-Curso4>

**MUDANÇAS NECESSÁRIAS**

Antes mesmo de começar o projeto, mudar dois pequenos equívocos

1 - Em ScreenSound.API/Appsettings.Development.json - mudar o banco de dados de ScreenSoundV1 (que não existe) para ScreenSoundV0 (que é o usado nos cursos anteriores).

1. Em ScreenSound.Dados/Migrations/ Na migration PopularMusicas, há uma linha

migrationBuilder.Sql("update Musicas set ArtistaId = (select Id from Artistas where Nome = 'Djavan')");

Comentar ela ou excluir.

Esse trecho da migration usa uma coluna que ainda nem foi criada (ArtistaId), cuja criação vai ser feita somente na migration posterior.

**INSTALAÇÃO E CONFIGURAÇÃO DO IDENTITY**

Para que seja possível ter controle de usuários, autenticação e autorização, usaremos o identity.

Instação do Identity no projeto ScreenSound.Shared.Dados

<PackageReference Include="Microsoft.AspnetCore.Identity.EntityFrameworkCore" Version="7.0.13" />

Criação da pasta modelos em ScreenSound.Shared.Dados, com as classes PessoaComAcesso e PerfilDeAcesso

PerfilDeAcesso - um *role*

using Microsoft.AspNetCore.Identity;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ScreenSound.Shared.Dados.Modelos

{

public class PerfilDeAcesso : IdentityRole<int>

{

}

}

PessoaComAcesso

using Microsoft.AspNetCore.Identity;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ScreenSound.Shared.Dados.Modelos

{

public class PessoaComAcesso : IdentityUser<int>

{

}

}

Mudança em Banco/ScreenSoundContext para incluir classes PessoaComAcesso e PerfilDeAcesso

public class ScreenSoundContext: IdentityDbContext<PessoaComAcesso, PerfilDeAcesso, int>

.

.

**ENDPOINTS DA API PARA USUÁRIOS**

Para fazer uso do identity na interface do sweager, vamos primeiro injetar o serviço do identity em program.cs

ScreenSound.API/program.cs

.

.

//INJEÇÃO SERVIÇO IDENTITY PARA GESTÃO DE ENDPOINTS DE ACESSO

builder.Services

.AddIdentityApiEndpoints<PessoaComAcesso>()

.AddEntityFrameworkStores<ScreenSoundContext>();

.

.

.

//MAPEAMENTO DOS ENDPOINST DO IDENTITY - GESTÃO DE ACESSO

app.MapGroup("auth").MapIdentityApi<PessoaComAcesso>().WithTags("Autorização");

//MapGroup("auth") - TODAS ROTAS MAPEADAS COMEÇARÃO COM ESSE CAMINHO

//.WithTags("Autorização"); - ORGANIZAÇÃO. NO SWAGGER, APARECERÃO JUNTAS NESSA TAG

Com essas configurações definidas, na rota auth/register podemos registrar usuários. Registraremos dois:

{

"email": "luciano@mail.com",

"password": "Senha!123"

}

{

"email": "luciano2@mail.com",

"password": "Senha!123"

}

Para verificar se os usuários foram registrados, usamos a rota auth/login, com *useCookies* true

{

"email": "luciano2@mail.com",

"password": "Senha!123"

}

**DETERMINANDO AUTORIZAÇÃO PARA ACESSO AOS DEMAIS ENDPOINTS DA APLICAÇÃO**

Para exemplificar uma aplicação de controle de acesso a um endpoint específico, vamos estabelecer que, para todos endpoints de artistas, será necessário estar identificado na API, para acessar. Como fazer isso?

Injetar o serviço de autorização em program.cs

//INJEÇÃO SERVIÇO DE AUTORIZAÇÃO

builder.Services.AddAuthorization();

.

.

//VERIFICAR REQUISIÇÕES HTTPS ANTES DE USAR ENDPOINTS

app.UseAuthorization();

Feito isso, vamos aplicar o uso dessa autorização aos endpoints de artista, que estão todos em Endpoinsts/ArtistaExtension

public static void AddEndPointsArtistas(this WebApplication app)

{

//VARIÁVEL groupBuilder QUE AGRUPA TODAS ROTAS COMEÇANDO COM artistas

//JÁ REQUERINDO AUTORIZAÇÃO

//E ORGANIZANDO COM TAG artistas

var groupBuilder = app.MapGroup("artistas")

.RequireAuthorization()

.WithTags("artistas");

#region Endpoint Artistas

groupBuilder.MapGet("", ([FromServices] DAL<Artista> dal) =>

{

.

.

groupBuilder.MapGet("{nome}", ([FromServices] DAL<Artista> dal, string nome) =>

.

.

Agora, ao tentar executar um endpoint de artista, sem que seja feito um login antes, o resultado é um code 401 (não autorizado)