WEB API - UsuariosApi

**DOWNLOADS NECESSÁRIOS**

VisualStudio Community 2022

.NET 6.0.402

**INÍCIO DO PROJETO - ESTRUTURA BASE**

Models->Usuario

Data->Dtos->CreateUsuarioDto

Data->UsuarioDbContext

Profile->UsuarioProfile

**Baixar**->Ferramentas->Gerenciador de Pacotes do NuGet->Gerenciar Pacotes NuGet para Solução->

AutoMapper.Extensions.Microsoft.DependencyInjection (v.12.0.0)

Controllers->UsuarioController

**CONFIGURAÇÃO IDENTITY E BD**

**Baixar**->Ferramentas->Gerenciador de Pacotes do NuGet->Gerenciar Pacotes NuGet para Solução->

Microsoft.AspNetCore.Identity.EntityFrameworkCore (v.6.0.14)

Microsoft.EntityFrameworkCore.Tools (v.6.0.14)

Microsoft.Extensions.Identity.Stores (v.6.0.14) MUDAR NOTEBOOK

Pomelo.EntityFrameworkCore.MySql (v.6.0.2)

Program-> linha 7 abaixo

//VARIÁVEL PARA DEFINIÇÕES ABAIXO

var connectionString = builder.Configuration.GetConnectionString("UsuarioConnection");

builder.Services.AddDbContext<UsuarioDbContext>(opts =>

opts.UseMySql(connectionString,

ServerVersion.AutoDetect(connectionString)));

//ADICIONANDO Identity AO PROJETO - RELACIONADO COM USUARIO E SEU PAPEL

builder.Services

.AddIdentity<Usuario, IdentityRole>()

.AddEntityFrameworkStores<UsuarioDbContext>() //IDENTITY CUIDA DA CONEXÃO COM BD ATRAVÉS DE UsuarioDbContext

.AddDefaultTokenProviders();//AUTENTICAÇÃO

appsettings.json-> linha 8 abaixo

"ConnectionStrings": {

"UsuarioConnection" : "server=localhost;database=usuariodb;user=root;password=root"

}

Aplicar mudanças a BD

Ferramentas-> Gerenciador de pacotes do Nuget->Console do Gerenciador de Pacotes

Add-Migration “Criando Usuarios”

Update-Database

UsuarioController->AdicionaUsuario

using AutoMapper;

using Microsoft.AspNetCore.Identity;

using Microsoft.AspNetCore.Mvc;

using UsuariosApi.Data.Dtos;

using UsuariosApi.Models;

namespace UsuariosApi.Controllers

{

[ApiController]

[Route("[Controller]")]

public class UsuarioController : ControllerBase

{

//PROPRIEDADES

private IMapper \_mapper; //AutoMap

private UserManager<Usuario> \_userManager;//GERENCIA MÉTODOS DO IDENTITY PARA USUÁRIO

//CONSTRUTOR

public UsuarioController(IMapper mapper, UserManager<Usuario> userManager)

{

\_mapper = mapper;

\_userManager = userManager;

}

//DEMAIS MÉTODOS

/// <summary> MÉTODO QUE ADICIONA UM USUÁRIO AO BD </summary>

/// <param name="dto">OBJETO DTO QUE RECEBE AS INFORMAÇÕES DE CRIAÇÃO</param>

/// <returns> Task<IActionResult> </returns>

/// <response code="201"> Em caso de inserção bem sucedida </response>

[HttpPost]

public async Task<IActionResult> CadastraUsuario(CreateUsuarioDto dto)

{ //async Task - RETORNO DE MÉTODO await É TIPO async Task<>

Usuario usuario = \_mapper.Map<Usuario>(dto);

//CRIAÇÃO USUÁRIO IDENTITY (USUÁRIO E SENHA)

//SENHA VEM DO OBJETO dto NESSE CASO, JÁ QUE OBJETO usuario NÃO TEM CAMPO SENHA

//await - INDICA QUE O PROGRAMA AGUARDA O USUÁRIO SER CRIADO PARA SALVAR EM resultado

IdentityResult resultado = await \_userManager.CreateAsync(usuario, dto.Password);

//EM CASO DE SUCESSO

if(resultado.Succeeded) return Ok("Usuário cadastrado!"); //201 CODE

//EM CASO DE FALHA

throw new ApplicationException("Falha ao cadastrar usuário");

}

}

}

Program->Configurar AutoMapper

//AUTOMAPPER

builder.Services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());

Tabela de usuários acessível pelo WorkBench em BD usuarioDB, tabela aspnetusers

Método de adicionar usuário acessível via Postman url

Até o momento: url por https não funciona.

\*Lembrar que: O Identity traz mecanismos de segurança - senha fraca, campo de confirmação de senha inválido,....

**ANOTAÇÕES**

//throw new NotImplementedException();