Table of Contents

[Current Resources with Changes 2](#_Toc114357727)

[/Core/DB/UserSeedData.json 2](#_Toc114357728)

[/Core/Entities/AppUser.cs 2](#_Toc114357729)

[/Core/Dto/UserDto.cs 2](#_Toc114357730)

[/Core/Repositories 2](#_Toc114357731)

[IUsersRepository.cs 2](#_Toc114357732)

[UsersRepository.cs 2](#_Toc114357733)

[/Core/BusinessLogic 3](#_Toc114357734)

[IUsersBusinessLogic 3](#_Toc114357735)

[UsersBusinessLogic 3](#_Toc114357736)

[UsersController.cs 3](#_Toc114357737)

[EF Handling 3](#_Toc114357738)

[Create Migrations and Update Database 3](#_Toc114357739)

[Drop Database 3](#_Toc114357740)

[Restart the App 4](#_Toc114357741)

# Current Resources with Changes

1. /Core/DB/UserSeedData.json
2. /Core/Entities/AppUser.cs
3. /Core/Dto/UserDto.cs
4. /Core/Repositories/UserRepository
5. /Core/BusinessLogic/UsersBusinessLogic
6. UsersController

# /Core/DB/UserSeedData.json

Added new property Guid and populated the guid as well for each user.

# /Core/Entities/AppUser.cs

Added indexing and new property Guid. Here are the changes

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using Microsoft.EntityFrameworkCore;

namespace MSC.Api.Core.Entities;

[Index(nameof(GuId))]

[Index(nameof(UserName))]

public class AppUser

{

    public int Id { get; set; }

    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]

    public Guid GuId { get; set; }

# /Core/Dto/UserDto.cs

New property added

public Guid GuId { get; set; }

# /Core/Repositories

## IUsersRepository.cs

New definition

    Task<UserDto> GetUserByGuidAsync(Guid id);

## UsersRepository.cs

Implement the new method

    public async Task<UserDto> GetUserByGuidAsync(Guid id)

    {

        //var user = await \_context.Users.FindAsync(id);

        //add photos as eager loading

        //var user = await \_context.Users.Include(p => p.Photos).SingleOrDefaultAsync(x => x.Id == id);

        //return user;

        //using automapper queryable extensions

        var user = await \_context.Users

                    .Where(x => x.GuId == id)

                    .ProjectTo<UserDto>(\_mapper.ConfigurationProvider)

                    .AsSplitQuery()

                    .AsNoTracking()

                    .SingleOrDefaultAsync();

        return user;

    }

# /Core/BusinessLogic

## IUsersBusinessLogic

Defined new method

    Task<UserDto> GetUserByGuidAsync(Guid id);

## UsersBusinessLogic

Implement the new method

    public async Task<UserDto> GetUserByGuidAsync(Guid id)

    {

        var user = await \_usersRepo.GetUserByGuidAsync(id);

        if (user == null) return null;

        //var userDto = new UserDto { Id = user.Id, UserName = user.UserName };

        //var userDto = \_mapper.Map<UserDto>(user);

        //return userDto;

        return user;

    }

# UsersController.cs

Create the new get method for the guid

    [HttpGet("{guid}/guid")]

    public async Task<ActionResult<UserDto>> GetUser(Guid guid)

    {

        var user = await \_usersBl.GetUserByGuidAsync(guid);

        if(user == null)

        {

            return NotFound($"No user found by guid {guid}");

        }

        return Ok(user);

    }

# EF Handling

## Create Migrations and Update Database

Issue the following two commands

* dotnet ef migrations add GuidAddedToUser -o Core/DB/Migrations
* dotnet ef database update

## Drop Database

Drop the database so that it is generated again

* dotnet ef database drop

|  |
| --- |
| Build started...  Build succeeded.  info: Microsoft.EntityFrameworkCore.Infrastructure[10403]  Entity Framework Core 6.0.6 initialized 'DataContext' using provider 'Microsoft.EntityFrameworkCore.Sqlite:6.0.6' with options: None  Are you sure you want to drop the database 'main' on server 'Core/DB/MySocialConnect.db'? (y/N)  y  info: Microsoft.EntityFrameworkCore.Infrastructure[10403]  Entity Framework Core 6.0.6 initialized 'DataContext' using provider 'Microsoft.EntityFrameworkCore.Sqlite:6.0.6' with options: None  Dropping database 'main' on server 'Core/DB/MySocialConnect.db'.  Successfully dropped database 'main'. |

# Restart the App

* dotnet watch run

It will restart the app and will also recreate the data.