Contents

[1. Models 2](#_Toc42340110)

[1.1. User 2](#_Toc42340111)

[2. DTO 2](#_Toc42340112)

[2.1. UserToRegister 2](#_Toc42340113)

[3. Repositories 3](#_Toc42340114)

[3.1. Auth Repository 3](#_Toc42340115)

[3.1.1. Interface: IAuthRepository 3](#_Toc42340116)

[3.1.2. Class: AuthRepository 3](#_Toc42340117)

## Models

## User

using System;

namespace MET.API.Models

{

    public class User

    {

        public int Id { get; set; }

        public string Username { get; set; }

        public byte[] PasswordHash { get; set; }

        public byte[] PasswordSalt { get; set; }

        public string EmailId { get; set; }

        [System.ComponentModel.DefaultValue("dev")]

        public DateTime CreatedOn { get; set; } = DateTime.Now;

        public DateTime LastActive { get; set; } = DateTime.Now;

    }

}

## DTO

## UserToRegister

using System.ComponentModel.DataAnnotations;

namespace MET.API.Dtos

{

    public class UserForRegisterDto

    {

        [Required]

        public string Username { get; set; }

        [Required]

        [StringLength(12, MinimumLength = 6, ErrorMessage = "Please Enter a Password with atleat 6 characters and maximum 12 characters")]

        public string Password { get; set; }

        [Required]

        [EmailAddress]

        public string Email { get; set; }

    }

}

## Repositories

## Auth Repository

## Interface: IAuthRepository

1. using System.Threading.Tasks;
2. using MET.API.Models;
3. namespace MET.API.Data
4. {
5. public interface IAuthRepository
6. {
7. Task<User> Register(User user, string password);
8. Task<User> Login(string username, string password);
9. Task<bool> UserExists(string username);
10. }
11. }

## Class: AuthRepository

using System.Threading.Tasks;

using MET.API.Data;

using MET.API.Dtos;

using MET.API.Models;

using Microsoft.AspNetCore.Mvc;

namespace MET.API.Controllers

{

    [Route("api/[controller]")]

    [ApiController]

    public class AuthController : ControllerBase

    {

        private readonly IAuthRepository \_repo;

        public AuthController(IAuthRepository repo)

        {

            \_repo = repo;

        }

        [HttpPost("register")]

        public async Task<ActionResult> Register(UserForRegisterDto userForrRegisterDto)

        {

            //  validate request

            userForrRegisterDto.Username = userForrRegisterDto.Username.ToLower();

            if(await \_repo.UserExists(userForrRegisterDto.Username))

            {

                return BadRequest("Username already exists");

            }

         var userToCreate = new User

         {

             Username = userForrRegisterDto.Username,

             EmailId = userForrRegisterDto.Email

         };

         var createdUser = await \_repo.Register(userToCreate, userForrRegisterDto.Password);

         return StatusCode(201);

        }

    }

}