|  |
| --- |
| using ChatApplication.Data; |
|  | using ChatApplication.Models; |
|  | using Microsoft.EntityFrameworkCore; |
|  | using Microsoft.Extensions.Logging; |
|  | using System; |
|  | using System.Collections.Generic; |
|  | using System.Threading.Tasks; |
|  |  |
|  | namespace ChatApplication.DataService |
|  | { |
|  | public class SQLLoginService : LoginService |
|  | { |
|  | private readonly ChatDBContext \_context; |
|  | private readonly ILogger \_Logger; |
|  | public SQLLoginService(ChatDBContext context, ILoggerFactory loggerFactory) |
|  | { |
|  | \_context = context; |
|  | \_Logger = loggerFactory.CreateLogger("UserRepository"); |
|  | } |
|  |  |
|  | public async Task<UserLogin> AddUserAsync(UserLogin info) |
|  | { |
|  | \_context.Add(info); |
|  | try |
|  | { |
|  | await \_context.SaveChangesAsync(); |
|  | } |
|  | catch (System.Exception exp) |
|  | { |
|  | \_Logger.LogError($"Error in {nameof(AddUserAsync)}: " + exp.Message); |
|  | } |
|  | return info; |
|  | } |
|  |  |
|  | public async Task<UserLogin> GetUserAsync(string name) |
|  | { |
|  | return await \_context.LoginData |
|  | .SingleOrDefaultAsync(c => c.name== name); |
|  | } |
|  |  |
|  | public async Task<List<UserLogin>> GetUsersAsync() |
|  | { |
|  | return await \_context.LoginData.ToListAsync(); |
|  | } |
|  |  |
|  | public async Task<bool> UpdateUserStatusAsync(UserLogin info) |
|  | { |
|  |  |
|  | \_context.LoginData.Attach(info); |
|  | \_context.Entry(info).State = EntityState.Modified; |
|  | try |
|  | { |
|  | return (await \_context.SaveChangesAsync() > 0 ? true : false); |
|  | } |
|  | catch (Exception exp) |
|  | { |
|  | \_Logger.LogError($"Error in {nameof(UpdateUserStatusAsync)}: " + exp.Message); |
|  | } |
|  | return false; |
|  | } |
|  | } |
|  | } |