|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Threading.Tasks; |
|  | using ChatApplication.Data; |
|  | using ChatApplication.Models; |
|  | using Microsoft.EntityFrameworkCore; |
|  | using Microsoft.Extensions.Logging; |
|  |  |
|  | namespace ChatApplication.DataService |
|  | { |
|  | public class SQLMessageService : MessageService |
|  | { |
|  | private readonly ChatDBContext \_context; |
|  | private readonly ILogger \_Logger; |
|  | public SQLMessageService(ChatDBContext context, ILoggerFactory loggerFactory) |
|  | { |
|  | \_context = context; |
|  | \_Logger = loggerFactory.CreateLogger("MessageRepository"); |
|  | } |
|  | public async Task<Messages> AddMsgAsync(Messages data) |
|  | { |
|  | \_context.Add(data); |
|  | try |
|  | { |
|  | await \_context.SaveChangesAsync(); |
|  | } |
|  | catch (System.Exception exp) |
|  | { |
|  | \_Logger.LogError($"Error in {nameof(AddMsgAsync)}: " + exp.Message); |
|  | } |
|  | return data; |
|  | } |
|  |  |
|  | public async Task<List<Messages>> GetMsgAsync() |
|  | { |
|  | return await \_context.ChatMessage.ToListAsync(); |
|  | } |
|  |  |
|  | public async Task<bool> UpdateReadStatusAsync(Messages data) |
|  | { |
|  | \_context.ChatMessage.Attach(data); |
|  | \_context.Entry(data).State = EntityState.Modified; |
|  | try |
|  | { |
|  | return (await \_context.SaveChangesAsync() > 0 ? true : false); |
|  | } |
|  | catch (Exception exp) |
|  | { |
|  | \_Logger.LogError($"Error in {nameof(UpdateReadStatusAsync)}: " + exp.Message); |
|  | } |
|  | return false; |
|  | } |
|  | } |
|  | } |