|  |
| --- |
| using System; |
|  | using System.Collections.Generic; |
|  | using System.Threading.Tasks; |
|  | using ChatApplication.DataService; |
|  | using ChatApplication.Models; |
|  | using Microsoft.AspNetCore.Mvc; |
|  | using Microsoft.Extensions.Logging; |
|  |  |
|  | namespace ChatApplication.Apis |
|  | { |
|  | [Produces("application/json")] |
|  | [Route("api/message")] |
|  | public class MessageController : Controller |
|  | { |
|  | private MessageService \_msgservice; |
|  | private ILogger \_Logger; |
|  |  |
|  | public MessageController(MessageService msgservice, ILoggerFactory loggerFactory) |
|  | { |
|  | \_msgservice = msgservice; |
|  | \_Logger = loggerFactory.CreateLogger(nameof(MessageController)); |
|  | } |
|  | [HttpGet] |
|  | [ProducesResponseType(typeof(List<Messages>), 200)] |
|  | [ProducesResponseType(typeof(messageResponse), 400)] |
|  | public async Task<ActionResult> Messages() |
|  | { |
|  | try |
|  | { |
|  | var msgs = await \_msgservice.GetMsgAsync(); |
|  | return Ok(msgs); |
|  | } |
|  | catch (Exception exp) |
|  | { |
|  | \_Logger.LogError(exp.Message); |
|  | return BadRequest(new messageResponse { Status = false }); |
|  | } |
|  | } |
|  | [HttpPost] |
|  | [ProducesResponseType(typeof(messageResponse), 201)] |
|  | [ProducesResponseType(typeof(messageResponse), 400)] |
|  | public async Task<ActionResult> AddMsg([FromBody]Messages data) |
|  | { |
|  | if (!ModelState.IsValid) |
|  | { |
|  | return BadRequest(new messageResponse { Status = false, ModelState = ModelState }); |
|  | } |
|  | try |
|  | { |
|  | var newmsg = await \_msgservice.AddMsgAsync(data); |
|  | if (newmsg == null) |
|  | { |
|  | return BadRequest(new messageResponse { Status = false }); |
|  | } |
|  | return CreatedAtRoute("GetMessageRoute", new { id = newmsg.id }, newmsg); |
|  | } |
|  | catch (Exception exp) |
|  | { |
|  | \_Logger.LogError(exp.Message); |
|  | return BadRequest(new messageResponse { Status = false }); |
|  | } |
|  | } |
|  | [HttpPut("{id}")] |
|  | [ProducesResponseType(typeof(messageResponse), 200)] |
|  | [ProducesResponseType(typeof(messageResponse), 400)] |
|  | public async Task<ActionResult> UpdateRead(int id, [FromBody]Messages data) |
|  | { |
|  | if (!ModelState.IsValid) |
|  | { |
|  | return BadRequest(new messageResponse { Status = false, ModelState = ModelState }); |
|  | } |
|  |  |
|  | try |
|  | { |
|  | var status = await \_msgservice.UpdateReadStatusAsync(data); |
|  | if (!status) |
|  | { |
|  | return BadRequest(new messageResponse { Status = false }); |
|  | } |
|  | return Ok(new messageResponse { Status = true, messages = data }); |
|  | } |
|  | catch (Exception exp) |
|  | { |
|  | \_Logger.LogError(exp.Message); |
|  | return BadRequest(new messageResponse { Status = false }); |
|  | } |
|  | } |
|  | } |
|  | } |