

```

using Meditech.App.Crm.Services;
using Meditech.App.Crm.ViewModels;
using Meditech.Core.Auth.Authorization;
using Meditech.Core.Auth.Entities;
using Meditech.Lib.Share.CommonClass.Paging;
using Microsoft.AspNetCore.Mvc;

namespace Meditech.App.Crm.Controllers
{
    [MediAuthorize]
    [ApiController]
    [Route("{culture:culture}/api/v{version:apiVersion}/{controller}")]
    [ApiVersion("1.0")]
    public class CommentController : ControllerBase
    {
        private readonly ILogger<CommentController> _logger;
        private readonly ICommentService _commentService;
        public CommentController(ILogger<CommentController> logger,
                                ICommentService commentService)
        {
            _logger = logger;
            _commentService = commentService;
        }

        [HttpPost]
        [Route("search-grid")]
        public IActionResult Search(CommentSearchViewModel model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _commentService.Search(model);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"Search error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpPost]
        [Route("mobile/search")]
        public IActionResult SearchCommunityQA(PaginationQueryModel<CommentSearchQueryData> model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _commentService.SearchMobile(model, requestUser.Id);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"SearchCommunityQA error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpGet]
        [Route("get-by-id")]
        public IActionResult DetailMobile(Guid id)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _commentService.CommentDetail(id, requestUser.Id, false);
                return Ok(new { Data = data });
            }
        }
    }
}

```

```

        catch (Exception ex)
        {
            _logger.LogError(ex, $"CommemtById error. ");
            return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
        }
    }

    [HttpGet]
    [Route("mobile/get-by-id")]
    public IActionResult CommemtById(Guid id)
    {
        try
        {
            var requestUser = (BaseUser)Request.HttpContext.Items["User"];
            if (requestUser == null) return Forbid();
            var data = _commentService.CommentDetail(id, requestUser.Id, true);
            return Ok(new { Data = data });
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"CommemtById error. ");
            return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
        }
    }

    [HttpPost]
    public IActionResult AddNewComment(CreateCommentViewModel model)
    {
        try
        {
            var requestUser = (BaseUser)Request.HttpContext.Items["User"];
            if (requestUser == null) return Forbid();
            var data = _commentService.AddNewComment(model, requestUser.Id);
            if (data.IsOk) return Ok(new { Data = data });
            return StatusCode(600, data);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"AddNewComment error. ");
            return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
        }
    }

    [HttpPut]
    public IActionResult UpdateComment(UpdateCommentViewModel model)
    {
        try
        {
            var requestUser = (BaseUser)Request.HttpContext.Items["User"];
            if (requestUser == null) return Forbid();
            var data = _commentService.UpdateComment(model, requestUser.Id);
            if (data.IsOk) return Ok(new { Data = data });
            return StatusCode(600, data);
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"UpdateComment error. ");
            return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
        }
    }

    [HttpDelete]
    public IActionResult Delete(Guid id)
    {
        try
        {
            var requestUser = (BaseUser)Request.HttpContext.Items["User"];
            if (requestUser == null) return Forbid();
            var data = _commentService.DeleteComment(id, requestUser.Id);
            if (data.IsOk) return Ok(new { Data = data });
            return StatusCode(600, data);
        }
        catch (Exception ex)
        {

```

```

        _logger.LogError(ex, $"Delete error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

[HttpPost]
[Route("like")]
public IActionResult UpSertLikeComment(UpSertLikeCommentViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _commentService.UpSertLikeComment(model, requestUser.Id);
        if (data.IsOk) return Ok(new { Data = data });
        return StatusCode(600, data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"UpSertLikeComment error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

```