

```

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

namespace Meditech.App.CommonService.Controllers
{
    [MediAuthorize]
    [Route("api/[controller]")]
    [ApiController]
    public class CommonController : ControllerBase
    {
        private readonly ILogger<CommonController> _logger;
        private readonly ICommonService _commonService;
        private readonly IHistoryChangeService _historyChangeService;

        public CommonController(ILogger<CommonController> logger,
            ICommonService commonService,
            IHistoryChangeService historyChangeService)
        {
            _logger = logger;
            _commonService = commonService;
            _historyChangeService = historyChangeService;
        }

        [HttpGet]
        [Route("get-count-noti-signing")]
        public IActionResult GetCountTotalSigning()
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null)
                    return Forbid();

                var data = _commonService.GetCountTotalSigning(requestUser.Id);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetCountTotalSigning Error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }

        [HttpPost]
        [Route("search-history-change")]
        public IActionResult GetHistoryChange(PaginationQueryModel<SearchHistoryChangeViewModel> model)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null)
                    return Forbid();

                var data = _historyChangeService.GetHistoryChange(model);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetHistoryChange Error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }
    }
}

```

