

```

using Meditech.App.Crm.Services;
using Meditech.App.Crm.ViewModels;
using Meditech.Core.Auth.Authorization;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Newtonsoft.Json;

namespace Meditech.App.Crm.Controllers
{
    [ApiController]
    [Route("{culture:culture}/api/v{version:apiVersion}/{controller}")]
    [ApiVersion("1.0")]
    public class RatingController : ControllerBase
    {
        private readonly ILogger<RatingController> _logger;
        private readonly IRatingService _ratingService;
        public RatingController(ILogger<RatingController> logger,
                                IRatingService ratingService)
        {
            _logger = logger;
            _ratingService = ratingService;
        }
        [HttpGet]
        [Route("get-by-code")]
        public IActionResult GetByCode(string code)
        {
            try
            {
                var data = _ratingService.GetByExternalId(code);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetByCode error. Code={code}");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }
        [HttpPost]
        [Route("submit")]
        public IActionResult AddRating([FromBody] SubmitRatingViewModel model)
        {
            try
            {
                var ret = _ratingService.SubmitRating(model);
                if (ret.IsOk) return Ok();
                return StatusCode(600, ret);
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"AddRating error. Data={JsonConvert.SerializeObject(model)}");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }
        [HttpPost]
        [Route("internal-zns/{secret}")]
        [MediAllowAnonymous]
        public async Task<IActionResult> TestZns(string secret, [FromBody] ZnsMessageViewModel model)
        {
            try
            {
                if (secret != "123456Aa!#")
                {
                    return StatusCode(StatusCodes.Status403Forbidden);
                }
                var ret = await _ratingService.SendZaloZnsAsync(model);
                if (ret.IsOk) return Ok();
                return StatusCode(600, ret);
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"TestZns error");
            }
        }
    }
}

```

```
return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
```

```
}
```

```
}
```

```
}
```

```
}
```