

```

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

namespace Meditech.App.Crm.Controllers
{
    [ApiController]
    [Route("{culture:culture}/api/v{version:apiVersion}/{controller}")]
    [ApiVersion("1.0")]
    public class PartnerController : ControllerBase
    {
        private readonly ILogger<PartnerController> _logger;
        private readonly IPartnerService _partnerService;

        public PartnerController(ILogger<PartnerController> logger, IPartnerService partnerService)
        {
            _logger = logger;
            _partnerService = partnerService;
        }

        #region Get Partner
        [HttpPost]
        [Route("get-partners")]
        [MediAllowAnonymous]
        //[[MediAuthorize(new[] { Lib.Share.PermitValue.Read, PermitValue.ReadAll }, Lib.Share.ActionPermit.Crm_CongTacVien)]]
        public async Task<IActionResult> GetPartners(PartnerFilterModel filter)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var model = new List<PartnerViewModel>();
                model = await _partnerService.GetAll(filter, requestUser.Id, true);
                return Ok(new { Data = model });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetPartners error.");
                return StatusCode(600, new ReturnCode
                {
                    Code = ErrorCode.HisSystemError,
                    Description = $"Lỗi hệ thống. Error={ex.Message}"
                });
            }
        }

        [HttpGet]
        [Route("get-partner-by-id")]
        [MediAuthorize(new[] { Lib.Share.PermitValue.Read }, Lib.Share.ActionPermit.Crm_CongTacVien)]
        public IActionResult GetPartnerById(Guid Id)
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();

                var data = _partnerService.GetById(Id);
                return Ok(data);
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetPartnerById error.");
                return StatusCode(600, new ReturnCode
                {
                    Code = ErrorCode.HisSystemError,
                    Description = $"Lỗi hệ thống. Error={ex.Message}"
                });
            }
        }
    }
    #endregion

    #region Create Partner
    [HttpPost]
    [Route("create-partner")]
    [MediAuthorize(new[] { Lib.Share.PermitValue.Create }, Lib.Share.ActionPermit.Crm_CongTacVien)]

```

```

public IActionResult CreatePartner([FromBody] PartnerViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();

        var data = _partnerService.Create(model, requestUser.Id);
        if (data.IsOk == false) return StatusCode(600, data);
        return Ok(data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"CreatePartner error. ");
        return StatusCode(600, new ReturnCode
        {
            Code = ErrorCode.HisSystemError,
            Description = $"Lỗi hệ thống. Error={ex.Message}"
        });
    }
}
#endregion

#region Update Partner

[HttpPut]
[Route("update-partner")]
[MediAuthorize(new[] { Lib.Share.PermitValue.Update }, Lib.Share.ActionPermit.Crm_CongTacVien)]
public IActionResult UpdatePartner([FromBody] PartnerViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();

        var data = _partnerService.Update(model, requestUser.Id);
        if (data.IsOk == false) return StatusCode(600, data);
        return Ok(data);
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"UpdatePartner error. ");
        return StatusCode(600, new ReturnCode
        {
            Code = ErrorCode.HisSystemError,
            Description = $"Lỗi hệ thống. Error={ex.Message}"
        });
    }
}
#endregion

#region Delete Partner
[HttpDelete]
[Route("delete-partner")]
[MediAuthorize(new[] { Lib.Share.PermitValue.Delete }, Lib.Share.ActionPermit.Crm_CongTacVien)]
public IActionResult DeletePartner(Guid id)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();

        var data = _partnerService.Delete(id);
        if (data.IsOk == false) return StatusCode(600, data);
        return Ok();
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"DeletePartner error. ");
        return StatusCode(600, new ReturnCode
        {
            Code = ErrorCode.HisSystemError,
            Description = $"Lỗi hệ thống. Error={ex.Message}"
        });
    }
}
#endregion
}
}

```

