

```
using Meditech.App.Crm.Services;
using Meditech.App.Crm.ViewModels;
using Meditech.Core.Auth.Entities;
using Microsoft.AspNetCore.Mvc;
```

```
namespace Meditech.App.Crm.Controllers
{
```

```
    [ApiController]
    [Route("{culture:culture}/api/v{version:apiVersion}/{controller}")]
    [ApiVersion("1.0")]
```

```
    public class SettingController : ControllerBase
    {
```

```
        private readonly ILogger<SettingController> _logger;
        private readonly ISettingService _settingService;
        public SettingController(ILogger<SettingController> logger,
                                ISettingService settingService)
```

```
        {
            _logger = logger;
            _settingService = settingService;
        }
```

```
        [HttpPost]
        [Route("banner/search-grid")]
        public IActionResult GetAllBanner(SearchGridBannerViewModel model)
```

```
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _settingService.GetAllBanner(model);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetAllBanner error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }
```

```
        [HttpGet]
        [Route("banner/mobile")]
        public IActionResult GetAllBannerMobile()
```

```
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _settingService.GetAllBannerMobile();
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {
                _logger.LogError(ex, $"GetBannerById error. ");
                return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
            }
        }
```

```
        [HttpGet]
        [Route("banner")]
        public IActionResult GetBannerById(Guid Id)
```

```
        {
            try
            {
                var requestUser = (BaseUser)Request.HttpContext.Items["User"];
                if (requestUser == null) return Forbid();
                var data = _settingService.GetBannerById(Id);
                return Ok(new { Data = data });
            }
            catch (Exception ex)
            {

```

```

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

[HttpPost]
[Route("banner")]
public IActionResult AddNewAllBanner(BannerSettingViewModel model)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _settingService.AddNewBanner(model);
        return Ok(new { Data = data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetAllBanner error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

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

[HttpDelete]
[Route("banner")]
public IActionResult DeleteBanner(Guid Id)
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _settingService.DeleteBanner(Id);
        return Ok(new { Data = data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"DeleteBanner error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

#region "SettingAll"
[HttpGet]
[Route("get-all")]
public IActionResult GetAllSetting()
{
    try
    {
        var requestUser = (BaseUser)Request.HttpContext.Items["User"];
        if (requestUser == null) return Forbid();
        var data = _settingService.GetAllSetting();
        return Ok(new { Data = data });
    }
    catch (Exception ex)
    {
        _logger.LogError(ex, $"GetAllSetting error. ");
        return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
    }
}

```

```

    }
    [HttpPut]
    [Route("update")]
    public IActionResult UpdateSetting([FromBody] UpdateSettingViewModel model)
    {
        try
        {
            var requestUser = (BaseUser)Request.HttpContext.Items["User"];
            if (requestUser == null) return Forbid();
            var data = _settingService.UpdateSetting(model);
            return Ok(new { Data = data });
        }
        catch (Exception ex)
        {
            _logger.LogError(ex, $"updateSetting error. ");
            return StatusCode(600, $"Lỗi hệ thống. Error={ex.Message}");
        }
    }
}
#endregion
}
}
}

```