Now that we have all the packages for Serilog installed, let's inject it inside our application and

let's provide the configuration for Serilog inside our application as well.

For that I will open program.cs file.

var builder = WebApplication.CreateBuilder(args);

// Configure Serilog for logging

var logger = new LoggerConfiguration()

.WriteTo.Console()

.MinimumLevel.Information()

.CreateLogger();

// clear default logging providers and add Serilog

builder.Logging.ClearProviders();

// Add Serilog as the logging provider

builder.Logging.AddSerilog(logger);

Program.cs file

//https://localhost:1234/api/regions

[Route("api/[controller]")]

[ApiController]

public class RegionsController : ControllerBase

{

private readonly NZWalksDbContext \_dbContext;

private readonly IRegionRepository \_regionRepository;

private readonly IMapper \_mapper;

private readonly ILogger<RegionsController> \_logger;

public RegionsController(NZWalksDbContext dbContext, IRegionRepository regionRepository,

IMapper mapper, ILogger<RegionsController> logger )

{

\_dbContext = dbContext;

\_regionRepository = regionRepository;

\_mapper = mapper;

\_logger = logger;

}

//Get All Regions

//Get: https://localhost:7050/api/regions

[HttpGet]

//[Authorize(Roles ="Reader")]

public async Task<IActionResult> GetAll()

{

\_logger.LogInformation("GetAll Regions API Method is called");

//calling repository to get all regions

var regionsDomain = await \_regionRepository.GetAllAsync();

//\_logger.LogInformation($"Finished getting all regions. Total Regions: {regionsDomain.Count()}");

\_logger.LogInformation($"Finished getting all regions request{JsonSerializer.Serialize(regionsDomain)}");

//Map Domain Models to DTOs using AutoMapper

var regionDTOs = \_mapper.Map<IEnumerable<RegionDTO>>(regionsDomain);//destination to source

//return Ok(regionsDomain);

return Ok(regionDTOs);

}