***// each of these import declarations should be commented –***

***// - what are we importing from each module?***

***// what do each of the imported functions do, in one phrase?***

***//***

using CPH\_IVT.Services.MongoDB.Init;

using CPH\_IVT.Services.MongoDB.Repository;

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using System;

using System.IO;

namespace CPH\_IVT

{

/// <summary>

/// ***no documentation yet. Explain this class’s operation***

/// </summary>

public class Driver

{

// Moved to the Async seeder to lower the number of params

// private static readonly string CurrentPath = Environment.CurrentDirectory;

// private static readonly string DataDirectory = Path.GetFullPath(Path.Combine(CurrentPath, "..", "..", "data", "indicators"));

// ***all params should be documented inline, rather than just referenced***

public static void Main(string[] args)

{

var host = CreateHostBuilder(args).Build(); ***// what is the host builder doing for us?***

SeedData(host); ***// why are we seeding it with data?***

host.Run(); ***// what does running the host accomplish?***

}

// ***should also be documented.***

// ***returns an interface to a host builder, where the web host defaults are defined using the Startup object?***

// ***what are the args??***

// ***all params should be documented inline, rather than just referenced***

// ***where are Host objects defined? In Microsoft.AspNetCore.Hosting?***

// ***what do Host objects provide?***

// ***where is Startup defined?***

//

public static IHostBuilder CreateHostBuilder(string[] args)

=> Host.CreateDefaultBuilder(args).ConfigureWebHostDefaults(webBuilder => { webBuilder.UseStartup<Startup>(); });

// ***should also be documented***

// ***what does it mean to create a scope?***

// ***what services do a ServiceProvider provide?***

// ***the code initializes a lot of variables, but to what end? Where do these variables live?***

// ***we ought to have a description for each of these variables***

// ***also, why not “just” write out Repository in the various names***

//

public static void SeedData(IHost host)

{

using var scope = host.Services.CreateScope();

var services = scope.ServiceProvider;

try

{

var initializer = services.GetRequiredService<InitializerAsync>();

var censusRegionRepo = services.GetRequiredService<ICensusRegionRepository>();

var censusDivisionRepo = services.GetRequiredService<ICensusDivisionRepository>();

var stateRepo = services.GetRequiredService<IStateRepository>();

var countyRepo = services.GetRequiredService<ICountyRepository>();

var healthIndicatorRepo = services.GetRequiredService<IHealthIndicatorRepository>();

// ***clean up the language! This is f\*ing unprofessional***

//

// Database seeder

// initializer.FuckingSendIt(healthIndicatorRepo, countyRepo, stateRepo, censusDivisionRepo, censusRegionRepo);

}

// ***can we do better with regard to logging an error message?***

// ***do we have any idea at this point as to which of the calls to GetRequiredService might have failed?***

catch (Exception e)

{

var logger = services.GetRequiredService<ILogger<Driver>>();

logger.LogError(e.Message);

}

}

}

}