***// censusRegion code is identical, up to substituting CensusRegion for CensusDivision***

***// customRegion code is identical, up to substituting CustomRegion for CensusDivision***

***// county code is identical, up to substituting County for CensusDivision***

***// health indicator code is identical, up to substituting HealthIndicator for CensusDivision***

***// state code is identical, up to substituting State for CensusDivision***

***// could these codes be collapsed?***

***// 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.Models;

using MongoDB.Driver;

namespace CPH\_IVT.Services.MongoDB.Context

{

/// <summary>

/// Provides a mechanism for NoSQL communication with <see cref="CensusDivision"/> documents.

/// </summary>

public interface ICensusDivisionContext

{

/// <summary>

/// MongoDB-specific collection of <see cref="CensusDivision"/> documents.

/// </summary>

/// <seealso cref="IMongoCollection{TDocument}"/>

IMongoCollection<CensusDivision> CensusDivisions { get; }

}

}

***// 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.Models;

using Microsoft.Extensions.Options;

using MongoDB.Driver;

namespace CPH\_IVT.Services.MongoDB.Context

/// <summary>

/// Represents a realization of <see cref="ICensusDivisionContext"/>.

/// </summary>

public class CensusDivisionContext : ICensusDivisionContext

{

/// <summary>

/// <see cref="IMongoDatabase"/>

/// </summary>

private readonly IMongoDatabase \_db;

/// <summary>

/// Parameterized constructor.

/// </summary>

/// <param name="options"><see cref="IOptions{TOptions}"/></param>

/// <seealso cref="HealthDatabaseSettings"/>

public CensusDivisionContext(IOptions<HealthDatabaseSettings> options)

{

var client = new MongoClient(options.Value.ConnectionString);

\_db = client.GetDatabase(options.Value.DatabaseName);

}

/// <summary>

/// MongoDB-specific collection of <see cref="CensusDivision"/> documents.

/// </summary>

/// <seealso cref="IMongoCollection{TDocument}"/>

/// <seealso cref="IMongoDatabase.GetCollection{TDocument}(string, MongoCollectionSettings)"/>

public IMongoCollection<CensusDivision> CensusDivisions => \_db.GetCollection<CensusDivision>("Census Divisions");

}

}