Eliassen - MongoDbExtensions

Summary

The MongoDbExtensions from Eliassen are intended to simplify configuration and instancation of Mongo Database collections for use with .Net 7.0+.

This provides a common means to build, configure and use collections by conventions to reduce complexity for developers.

Usage

Declare and Register Collections

Using the `Eliassen.MongoDB.Extensions` just requires creating and interface with getter only properties per collection you which to register.

Example

```
//collection definition
public class UserCollection
{
    [Key] //you may either use the BsonId and Bson
    public string? UserId { get; set; }
    public string? EmailAddress { get; set; }
    public string? FirstName { get; set; }
    public string? LastName { get; set; }
    public bool Active { get; set; }
    public List<UserModuleCollection>? UserModules { get; set; }
    public DateTimeOffset? CreatedOn { get; set; }
}

// database definition
public interface ICoreMongoDatabase
{
    //this attribute may be used to explicitly declare the name for the related
collection in MongoDB
    [CollectionName("users")]
    IMongoCollection<UserCollection> Users { get; }

    // if the collection name is not provided the configure Json Property Naming
Policy will be used.
    // Default is camel case.  In this example it would be "persons"
    IMongoCollection<PersonCollection> Persons { get; }
}
```

```
//registration
//In your IOC registration method use the `TryAddMongoDatabase<>` extension method from the `Eliassen.MongoDb.Extensions` namespace.
//this will create a proxy class in your IoC container to access your mongodb

using Eliassen.MongoDB.Extensions;

namespace Nucleus.Core.Persistence;

public static class ServiceCollectionEx
{
    public static IServiceCollection AddCorePersistenceServices(this IServiceCollection services)
    {
        services.TryAddMongoDatabase<ICoreMongoDatabase>();
        return services;
    }
}
```