

Introduction to .NET – lab no. 4

Way of working: in pair with one colleague.

Summary:

Keywords: DbContext, EF Core, Repository pattern, LINQ, method syntax

Prerequisites

1. Check the extra info's.
2. Create a new .NET Core class library
 - a. Create a new class Product and expose the same properties (discussed in the previous labs)
 - b. Using attributes make the following adjustments:
 - i. Name mandatory, maximum 50 characters
 - ii. StartDate mandatory
 - iii. EndDate nullable
 - iv. Price mandatory
 - v. Vat mandatory
 - c. Create a ProductManagement class derived from DbContext (add the required libraries)
 - d. Create class Customer with the following properties : Id, Name, Address, PhoneNumber, Email.
 - e. Using Fluent API make the following adjustments (override OnCreating)
 - i. Name mandatory maxim 100 characters
 - ii. Address mandatory, maxim 300 characters
 - iii. PhoneNumber mandatory, regular expression for RO
 - iv. Email mandatory, regular expression
 - f. Add Product and Customer to the Context
 - g. Create 2 repositories for both classes and expose the following behaviour: Create/Update/Delete/GetById, GetAll. (working with the Context implement the proper methods)

Note : accesing the context class should be done by using "Poor man IoC principle".
 - h. For the ProductRepository expose a special behaviour: GetProductsByPrice
 - i. For the CustomerRepository expose a special behaviour: GetCustomerByEmail.
 - j. Remove code redundancies.

Extra info's

Hints

Attributes: With attributes we can specify different aspects database oriented: Not Null field, dimensions for nvarchar, custom types

[Required]

[StringLength(40)]

public string CustomerName { get; set; }

or:

[Column(TypeName = "money")]

public decimal? UnitPrice { get; set; }

Fluent API: It can be used:

- ▶ Combined with attributes
- ▶ As a replacement for attributes

[Required]

[StringLength(40)]

public string CompanyName { get; set; }

They could be deleted and replaced with this Fluent API statement in the Context class OnModelCreating method:

modelBuilder.Entity<Customer>()

.Property(customer=>customer.CompanyName)

.IsRequired()

.HasMaxLength(40);

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    if (!optionsBuilder.IsConfigured)
    {
        optionsBuilder.UseSqlServer(@"Server=(localdb)\mssqllocaldb;Database=YourDatabaseName;Trusted_Connection=True;");
    }
}
```

Note:

1. All exercises are mandatory.
2. You will receive your points at the end of the lab.
3. Working time: 80 minutes.

