Opgave: ConsoleApplikation

Du skal lave en ny applikation til opgaven med BowlingBanen (se DatabaseOgConsoleApp), Denne gang skal du lave koden dvs. modelklasserne først og ud fra dem lave Database tabellerne.

Baggrund: RazorPageNote2 Chapter 3: Entity Framework- Code First side 52-62 (let 62-75)

Lave et lille program, hvor du benytter EntityFramework

Til det skal du have model-klasser, samt selve ConsoleApp'en.

Trin 1: Model-klasse(r)

1.A Model klasser

Du skal lave et projekt **'Console Application (.Net 5.0)'**, hvor du implementerer de tre model-klasser i en mappe 'model':

Person: med properties Phone, PName, ShoeSize
BowlingAlley: med properties Number, OperationalState
Booking: med properties Id, BookingDate, BookingTime, PersonPhone,AlleyNo

Du skal nu anvende nogle annotations som gør det muligt at lave tabeller i en databasef.eks:

[Key] over properties der skal være primær nøgler

[Required] over properties der skal være udfyldte

[StringLength(50)] over string properties

1.B Installer NuGet pakker

I NuGet installer pakkerne:

- Microsoft.EntityFrameWorkCore
- Microsoft.EntityFrameWorkCore.SqlServer
- Microsoft.EntityFrameWorkCore.Tools

1.C Lav din egen DB-context

I mappen model lav en klasse **BowlingDBContext,** Som arver fra DBContext.

Klassen skal overskrive en konfigurerings metode:

```
protected override void OnConfiguring(DbContextOptionsBuilder options)
   {
      options.UseSqlServer(
         @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Bowling;" +
          "Integrated Security=True;Connect Timeout=30;Encrypt=False");
        }
```

(PS! Din connectionstring kan være lidt anderledes)

Samt tre properties til mapning mellem objekter og tabeller

```
public DbSet<Player> Players { get; set; }
public DbSet<BowlingLane> Lanes { get; set; }
public DbSet<booking> Bookigs { get; set; }
```

1.D Opret Tabellerne

Åben Package manageren (Tools -> NuGet Package Manager -> Package Manager Console):

I terminal vinduet (Package Manager Console) skriv de to kommandoer:

- Add-Migration
- Update-database

1.E Undersøg tabellerne der er oprettet

Kig vha. 'SQL Server Object Explorer' og find databasen **Bowling**, samt de tre tabeller.

Trin 2: Console Applikation

Du skal i projektet i første omgang lave CRUD på Person (tabellen).

Lav en klasse AWorker med en Start-metode og kald denne fra program-main metoden.

I projektet skal du lave et interface '**IPersonService**' med følgende 5 metoder:

```
private List<Person> GetAllPersons()

private Person GetPersonByPhoneNo(string phone)

private Person AddPerson(Person person)

private Person DeletePerson(string phone)

private Person UpdatePerson(string phone, Person person)
```

I projektet skal du lave en klasse '**PersonService**' der implementerer interfacet IPersonService ved brug af din klasse **BowlingDBContext**.

Du skal oprette et objekt af din PersonService klasse og kalde de forskellige metoder fra Start-metoden.

Trin 3: Tilføj Fremmed nøgler

3.A Ændre model-klasser

Dine tabeller er uden fremmed nøgler, dem skal du tilføje nu.

Eksempel Person og Booking hænger sammen (Booking har en fremmed-Nøgle til Person)
Når vi skal lave det med code-first skal vi tilbage og ændre i model-klasserne.

I klassen Person skal tilføjes en navigation property:

ICollection<Booking> Bookings { get; set;}

For at være helt sikker angiver vi fremmedNøglen eksplicit med en annotation,
[ForeignKey("PersonPhone ")]

I klassen Booking tilføjes ligeledes en property

Person Person { get; set; }

For at være helt sikker angiver vi fremmedNøglen eksplicit med en annotation,
dvs. over propertyen PersonPhone
[ForeignKey("Person")]

3.B Opdater Databasen

Endnu engang åbne Package Manager Console og skriv de to kommandoer:

- Add-Migration
- Update-database

3.C Afprøv

Se i tabellerne hvad der er sket.
Hvordan påvirker det din applikation?

Ekstra 1: Udvid din applikation med BowlingAlley og Booking