

## 5. Óra

### Feladat: (Layering)

#### Szükséges NuGet csomagok:

- *Microsoft.EntityFrameworkCore.Proxies*
- *Microsoft.EntityFrameworkCore.Tools*
- *Microsoft.EntityFrameworkCore.InMemory*
- *Newtonsoft.Json*

### DbContext:

1. Hozzunk létre egy új **Console Application**-t és töltsük le a fentebb felsorolt **NuGet** csomagokat.  
***FONTOS: Mindig a megfelelő verziót töltsük le! .NET 5-höz az 5-tel kezdődő, .NET 6-hoz a 6-tal kezdődő verziót kell letölteni!***
2. Töltsük le a kettő JSON file-t a GitHub repository-ból és másoljuk be a projektbe. **Visual Studio**-n belül állítsuk be a **fájl tulajdonságai** között a **Build Action**-t **“Content”** értékre, a **Copy to Output Directory**-t pedig **“Copy Always”** értékre.

### Models réteg:

#### **Worker** osztály:

1. Hozzunk létre egy **Worker** osztályt a JSON fájl alapján, valamint adjunk még hozzá egy publikus, virtuális, **Company** típusú tulajdonságot Company néven.
2. Az EyeColor tulajdonság legyen **EyeColorEnum** típusú.
3. A Registered tulajdonság legyen **DateTime** típusú.
4. A CompanyId tulajdonság rendelkezzen egy **ForeignKey** attribútummal, ahol adjuk át neki a **Company** osztályt.
5. Legyen egy **string** típusú Id tulajdonsága, amelynek az értékét konstruktorban állítsuk be a **GUID** segítségével. A tulajdonságot jelöljük hogy egy elsődleges kulcs legyen és növelje az adatbázis automatikusan az értékét.

#### **Company** osztály:

1. Hozzunk létre egy **Company** osztályt a JSON fájl alapján, valamint adjunk hozzá még egy publikus, virtuális, **ICollection<Worker>** típusú tulajdonságot Workers néven. *(Példányosítsuk a konstruktorban egy sima listával)*

2. A CountryId tulajdonság rendelkezzen egy **ForeignKey** attribútummal, ahol adjuk át neki a **Country** osztályt.
3. Hozzunk létre egy publikus, virtuális, **Country** típusú tulajdonságot Country néven.
4. Az Id tulajdonságát jelöljük hogy egy **elsődleges kulcs** attribútummal és növelje az adatbázis automatikusan.

#### **Country** osztály:

1. Hozzunk létre egy **Country** osztályt a JSON fájl alapján, valamint adjunk hozzá még egy publikus, virtuális, **ICollection<Company>** típusú tulajdonságot Companies néven. *(Példányosítsuk a konstruktorban egy sima listával)*
2. Az Id tulajdonságát jelöljük hogy egy **elsődleges kulcs** attribútummal és növelje az adatbázis automatikusan.

#### **Data réteg:**

##### **CompanyDbContext** osztály:

4. Készítsünk egy **CompanyDbContext** osztályt, amely száramazzon le a **DbContext** osztályból. Konstruktorban biztosítsuk hogy az adatbázis létrejön. Hozzunk létre három **DbSet** típusú tulajdonságot a tábláinknak. Valamint írjuk felül az **OnConfiguring** és **OnModelCreating** metódusait.
5. Az **OnConfiguring** metóduson belül **InMemory** adatbázist használjunk.
6. Az **OnModelCreating** metóduson belül olvassuk be a három JSON fájl tartalmát, ezek lesznek a seed adatok. Ehhez segítségül készítsünk egy privát, generikus **ParseJson** metódust, ami a fájl nevét várja bemenetként. *(Elég ha csak a metódus generikus, nem kell az egész osztálynak annak lennie)*

## Repository réteg:

### Repository osztályok:

1. Hozzunk létre minden létező modelnek egy repository osztályt, amelyben megvalósítjuk a kapcsolatot az adatbázissal. *Például: WorkerRepository*
2. Szükséges egy generikus **IRepository interface**, ami rendelkezzen az alábbi metódusokkal:
  - `void Create(T t);`
  - `T Get(string id);`
  - `IQueryable<T> GetAll();`
  - `void Delete(string id);`
3. Minden repository osztály valósítsa meg ezt az interfészt.

## Logic réteg:

### Logic osztályok:

1. Hozzunk létre minden létező modelnek egy logic osztályt, amelyben megvalósítjuk a kapcsolatot az Repository réteggel. *Például: WorkerLogic*
2. Szükséges egy generikus **ILogic interface**, ami rendelkezzen az alábbi metódusokkal:
  - `void Create(T t);`
  - `T Get(string id);`
  - `IQueryable<T> GetAll();`
  - `void Delete(string id);`
3. Minden logic osztály valósítsa meg ezt az interfészt. Figyeljünk, hogy a megfelelő típust adjuk át az interface-nek.

## Program.cs:

- Hozzunk létre a **Program.cs**-ben minden model logic osztályából egy példányt.
- Teszteljük le az összes metódust.