

# Oefening DataLayer met EF Core

In deze oefening gaan we een multilayer App maken in een solution en maken we de DataLayer met EF Core (Database first)

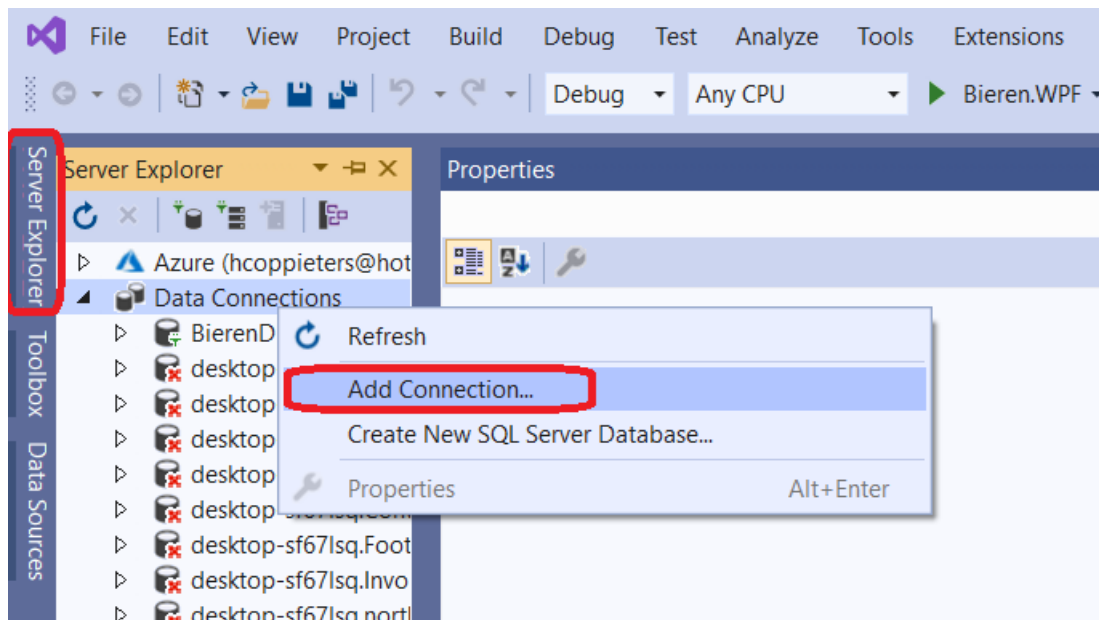
Download **Bieren\_start.zip** van GitHub (LES 4)

## Open de solution

Voeg een nieuw project toe binnen deze solution: Class Library (.NET Core) (reeds gedaan)

Nog te doen:

- Leg een **project Reference** van Bieren.WPF naar Bieren.DataLayer (niet omgekeerd!)
- **Installeer NuGet Packages** (in alle 2 de projecten):  
Microsoft.EntityFrameworkCore.SqlServer  
Microsoft.EntityFrameworkCore.Tools  
Microsoft.EntityFrameworkCore.Design
- Leg in de Server Explorer een DataConnectie naar het bestand **BierenDb.mdf** (in folder Data in project Bieren.DataLayer):



- We gaan een **rechtstreeks** connectie leggen naar een database **file** in plaats van te gaan via een SQL server instantie.
- **Klik op de button 'Change' in het Add-Connectie pop-up venster:**

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
Microsoft SQL Server (SqlClient) Change...

Server name:  
[Dropdown] Refresh

Log on to the server  
Authentication: Windows Authentication  
User name: [Text Box]  
Password: [Text Box]  
☐ Save my password

Connect to a database  
☒ Select or enter a database name:  
[Dropdown]  
☐ Attach a database file:  
[Text Box] Browse...  
Logical name:  
[Text Box]

Advanced...

Test Connection OK Cancel

- **Selecteer "Microsoft SQL Server Database File" uit de lijst en klik op OK button:**

Change Data Source

Data source:  
Microsoft Access Database File  
Microsoft ODBC Data Source  
**Microsoft SQL Server Database File**  
Oracle Database  
<other>

Description  
Use this selection to attach a database file to a local Microsoft SQL Server instance (including Microsoft SQL Express) using the .NET Framework Data Provider for SQL Server.

Data provider:  
.NET Framework Data Provider for S

☐ Always use this selection

OK Cancel

- Klik op **Browse..** en selecteer de BierenDb.mdf (in Folder Data onder Bieren.DataLayer) en klik daarna op OK

Add Connection

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:  
Microsoft SQL Server Database File (SqlClient) Change...

Database file name (new or existing):  
 Browse...

Log on to the server

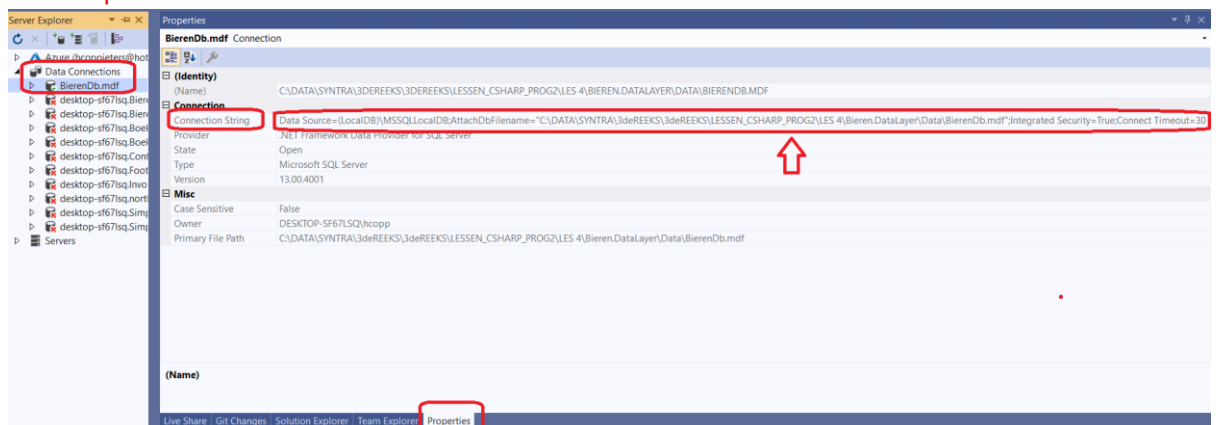
☒ Use Windows Authentication  
☐ Use SQL Server Authentication

User name:   
Password:   
☐ Save my password

Advanced...

Test Connection OK Cancel

- Vraag de properties van de nieuwe Dataconnectie op (vb via F4 toets) en kopieer de volledige connectionstring in notepad en vervang de dubbele quotes rond het pad door enkele quotes :

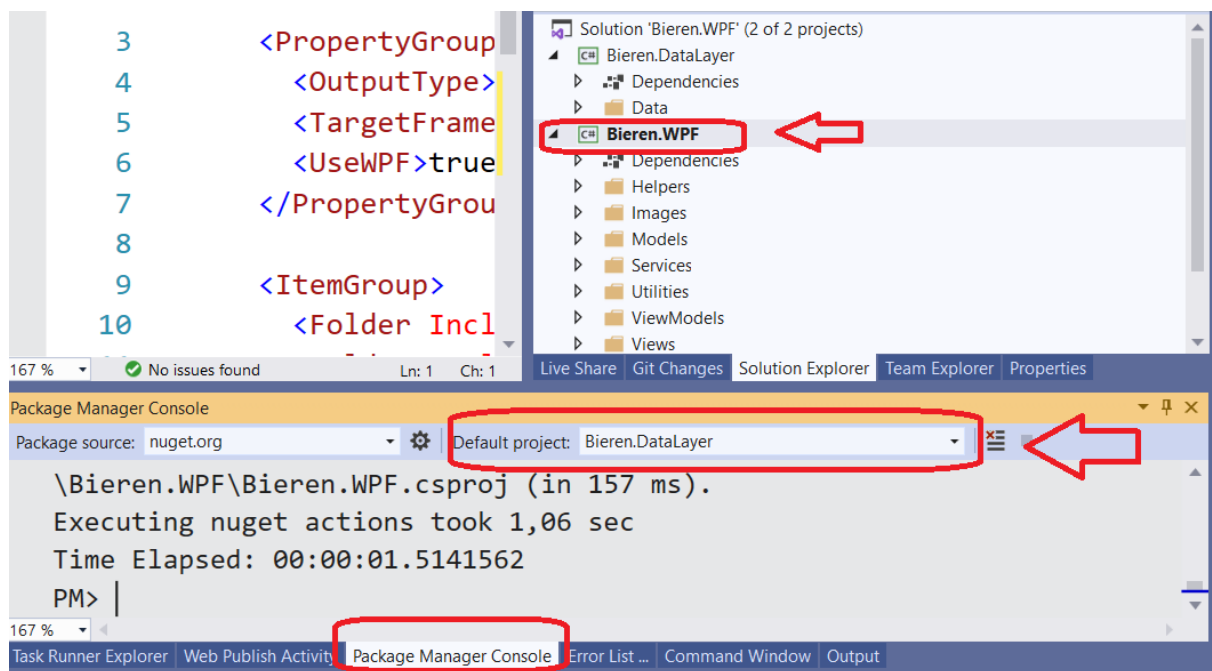


## Bv connectionstring:

"Data

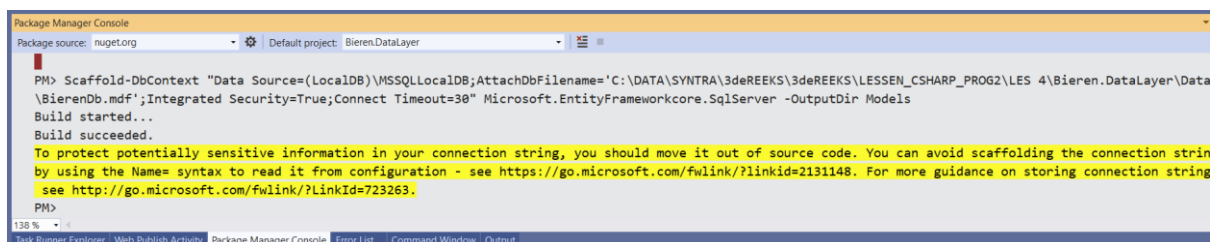
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\DATA\SYNTRA\DATA\BierenDb.MDF';Integrated Security=True;Connect Timeout=30"

- Open het Package Manager Console venster, run de volgende instructie (**Bieren.DataLayer moet als Default project staan !!**) zonder linebreaks of extra spaties
- Scaffold-DbContext **"..querystring..."** Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

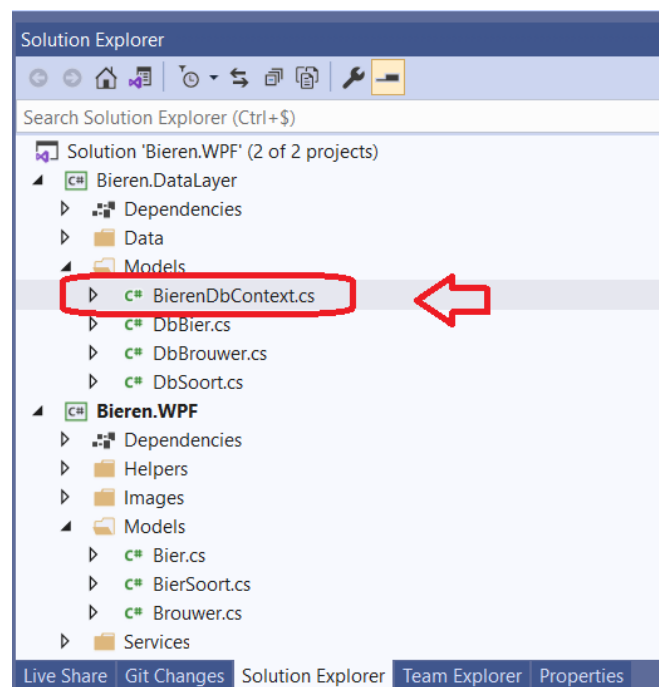
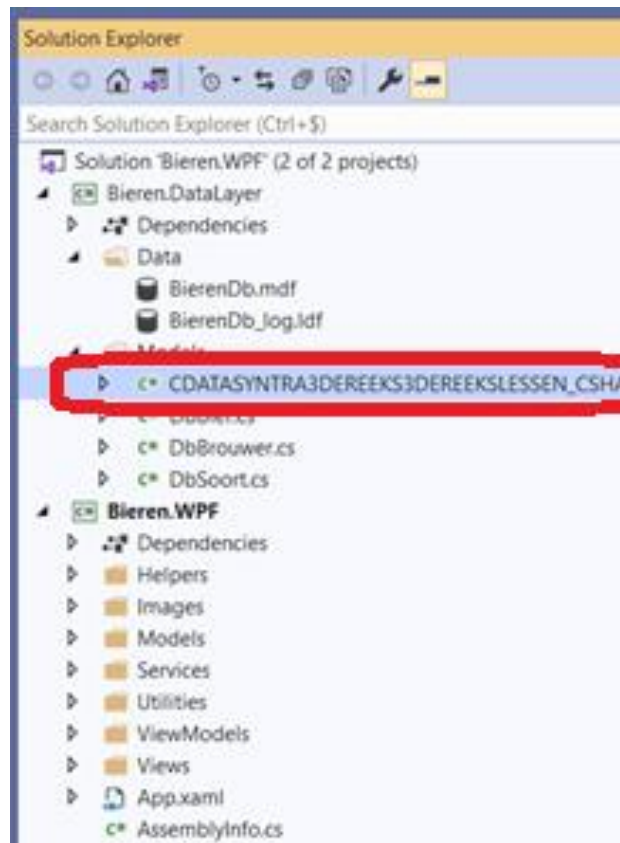


## Bv

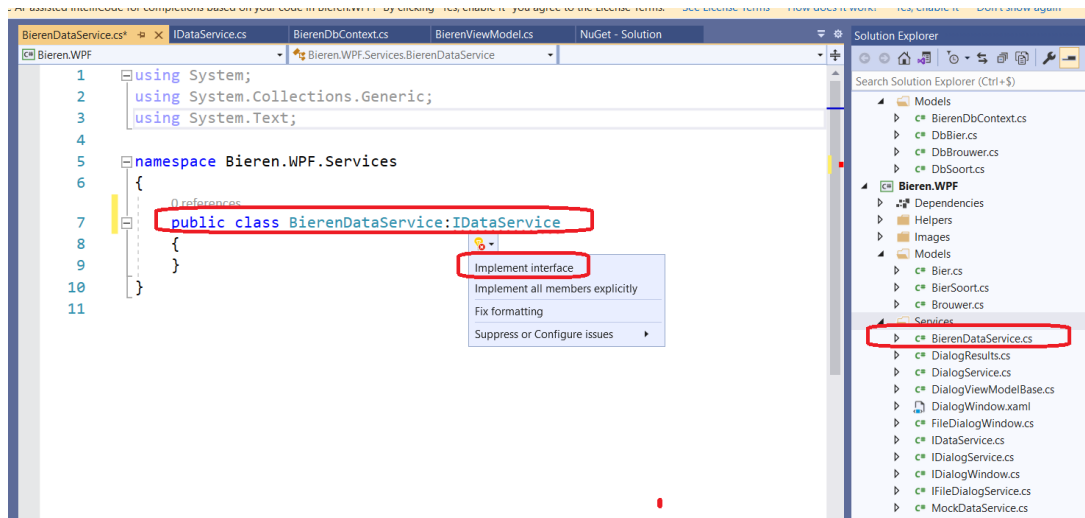
- Scaffold-DbContext **"Data**  
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename='C:\DATA\SYNTRA\3deREEKS\3deREEK  
S\LESSEN\_CSHARP\_PROG2\LES 4\Bieren.DataLayer\Data\BierenDb.mdf';Integrated  
Security=True;Connect Timeout=30" Microsoft.EntityFrameworkCore.SqlServer -OutputDir  
Models



- Wijzig de automatisch gegenereerde naam van de xxxxxxxContext class via de Solution Explorer naar **BierenDbContext.cs** :



- Voeg onder de **folder Services** in het project **Bieren.WPF** een nieuwe **class** toe met naam **BierenDataService** die de interface **IDataService** implementeert :



- Rechtsklik op **IDataService** in het codevenster en kies **'Implement interface'**:

Vervang de lijnen

```
throw new NotImplementedException();
```

door echte implementaties:

bv:

```
public IList<Bier> GeefAlleBieren()
{
    return DbBierenToBieren();
}
1 reference
private IList<Bier> DbBierenToBieren()
{
    IList<Bier> bieren = new List<Bier>();
    using (BierenDbContext db = new BierenDbContext())
    {
        var dbBieren = db.DbBiers.Include(b => b.BrouwerNrNavigation).Include(b => b.SoortNrNavigation);
        foreach (DbBier dbBier in dbBieren)
        {
            bieren.Add(DbBierToBier(dbBier));
        }
    }
    return (IList<Bier>)bieren;
}
2 references
private Bier DbBierToBier(DbBier dbBier)
{
    if (dbBier is null) return null;
    Bier bier = new Bier()
    {
        BierNr = dbBier.BierNr,
        Naam = dbBier.Naam,
        Alcohol = dbBier.Alcohol,
        Brouwer = DbBrouwerToBrouwer(dbBier.BrouwerNrNavigation),
        BierSoort = DbSoortToBierSoort(dbBier.SoortNrNavigation)
    };
    return bier;
}
```

```

private Brouwer DbBrouwerToBrouwer(DbBrouwer dbBrouwer)
{
    if (dbBrouwer is null) return null;
    return new Brouwer()
    {
        BrouwerNr = dbBrouwer.BrouwerNr,
        BrNaam = dbBrouwer.BrNaam,
        Straat = dbBrouwer.Adres,
        PostCode = dbBrouwer.PostCode,
        Gemeente = dbBrouwer.Gemeente,
        Omzet = dbBrouwer.Omzet
    };
}

```

5 references

```

public IList<BierSoort> GeefAlleBierSoorten()
{
    return DbSoortenToBierSoorten();
}

```

4 references

```

public IList<Brouwer> GeefAlleBrouwers()
{
    return DbBrouwersToBrouwers();
}

```

1 reference

```

private IList<Brouwer> DbBrouwersToBrouwers()
{
    IList<Brouwer> brouwers = new List<Brouwer>();
    using (BierenDbContext db = new BierenDbContext())
    {
        var dbBrouwers = db.DbBrouwers;
        foreach (DbBrouwer dbBrouwer in dbBrouwers)
        {
            brouwers.Add(DbBrouwerToBrouwer(dbBrouwer));
        }
    }
    return (IList<Brouwer>)brouwers;
}

```

Open `MainViewModel.cs` wijzig de Creatie van De dataservice in de constructor:

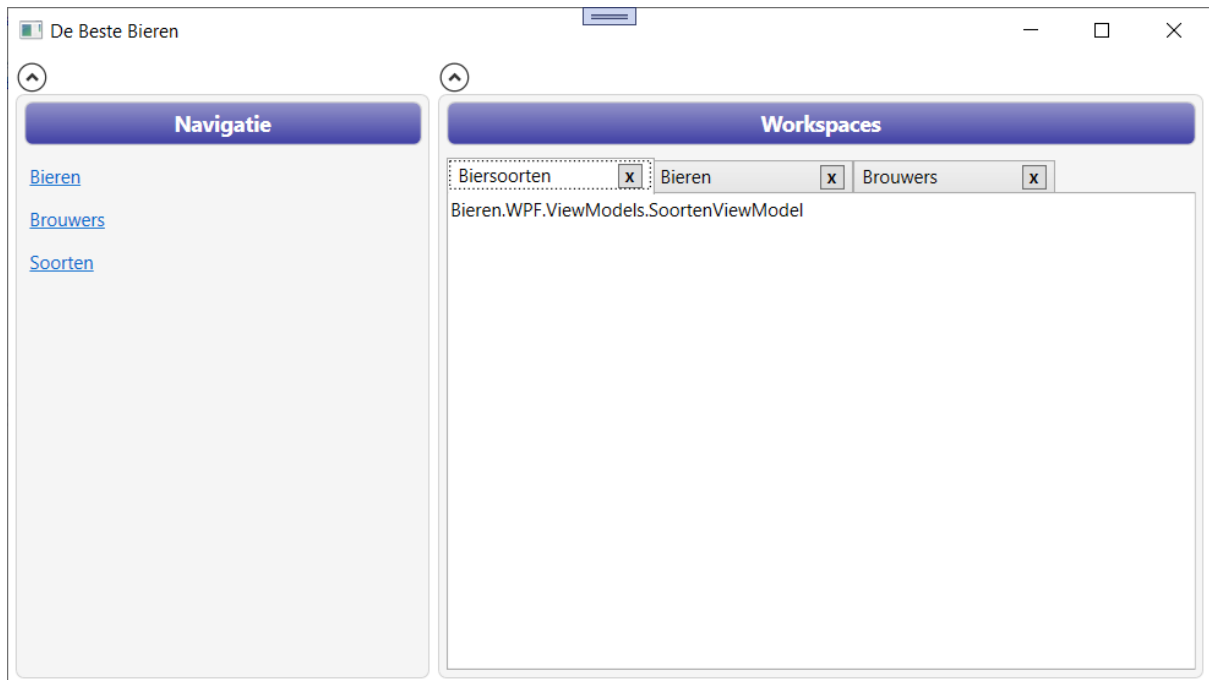
```

_dataService =new MockDataService(); door
_dataService =new BierenDataService();

```

Run de app en controleer of alles goed werkt en geïmplementeerd is

Zorg dat er op de Tab van de biersoorten een lijst van biersoorten verschijnt en de mogelijkheid om een biersoort toe te voegen, een geselecteerde biersoort aan te passen (enkel naam) en een geselecteerde biersoort te verwijderen.



Run de app en controleer of alles goed werkt en geïmplementeerd is