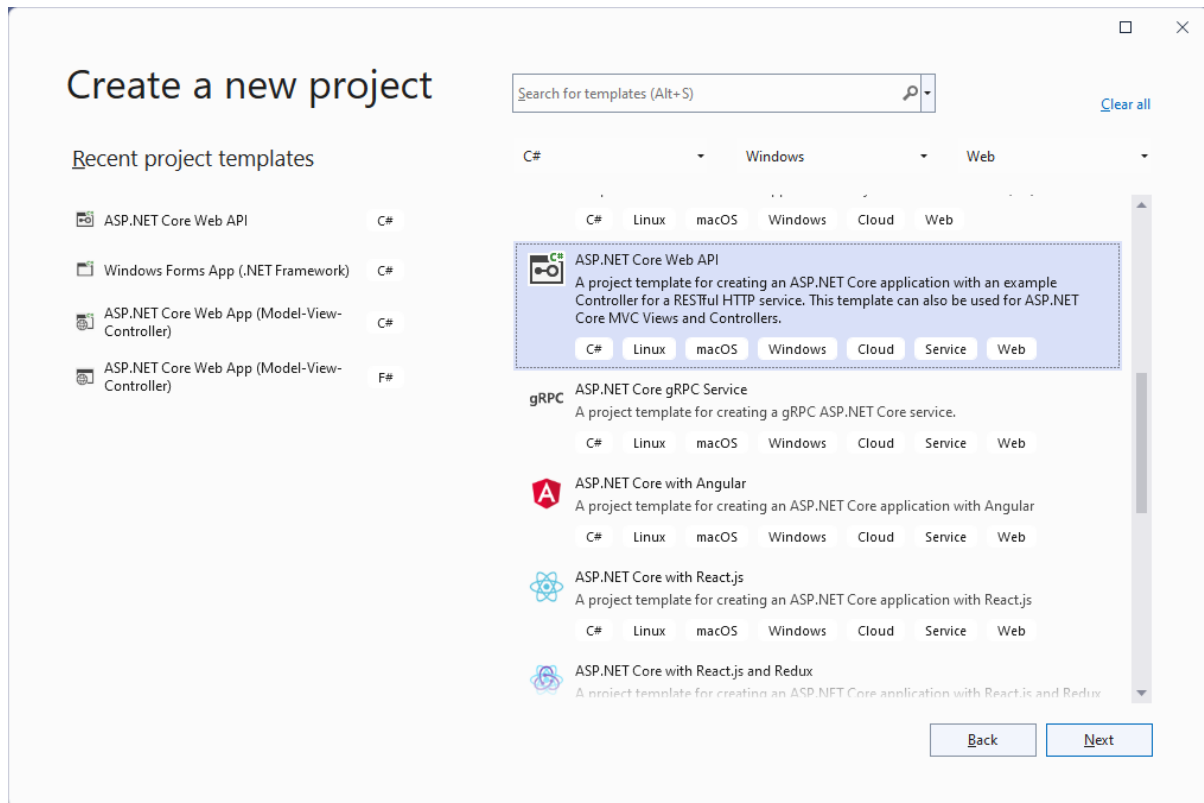


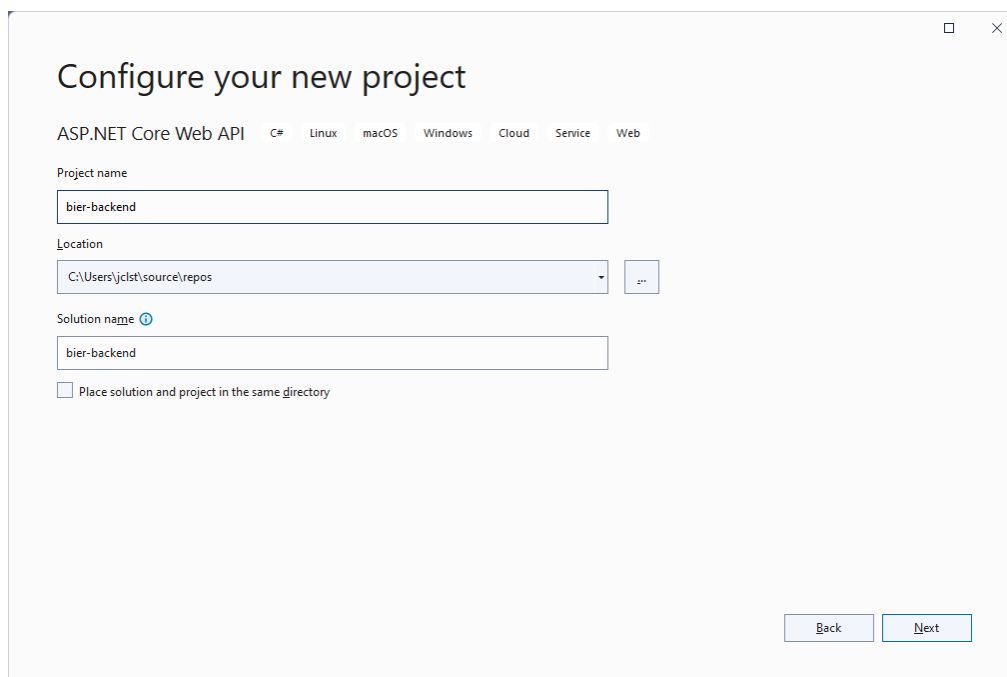
Een API BACKEND maken ZONDER authenticatie en MET SQLite Database (incl Swagger).

Benodigheden: Visual Studio 2022 (CE), DBeaver, en het BIER.SQL bestand geschikt voor SQLite.

START



Stap 1. Kies een ASP.NET Core Web API project.



Stap 2. geef je project een naam bv. bier-backend.

Stap 3. We kiezen 'latest greatest' bij Framework, we gaan nog geen gebruik maken van Authenticatie.

Additional information

ASP.NET Core Web API C# Linux macOS Windows Cloud Service Web

Framework ⓘ
[.NET 6.0 (Long-term support)]

Authentication type ⓘ
[None]

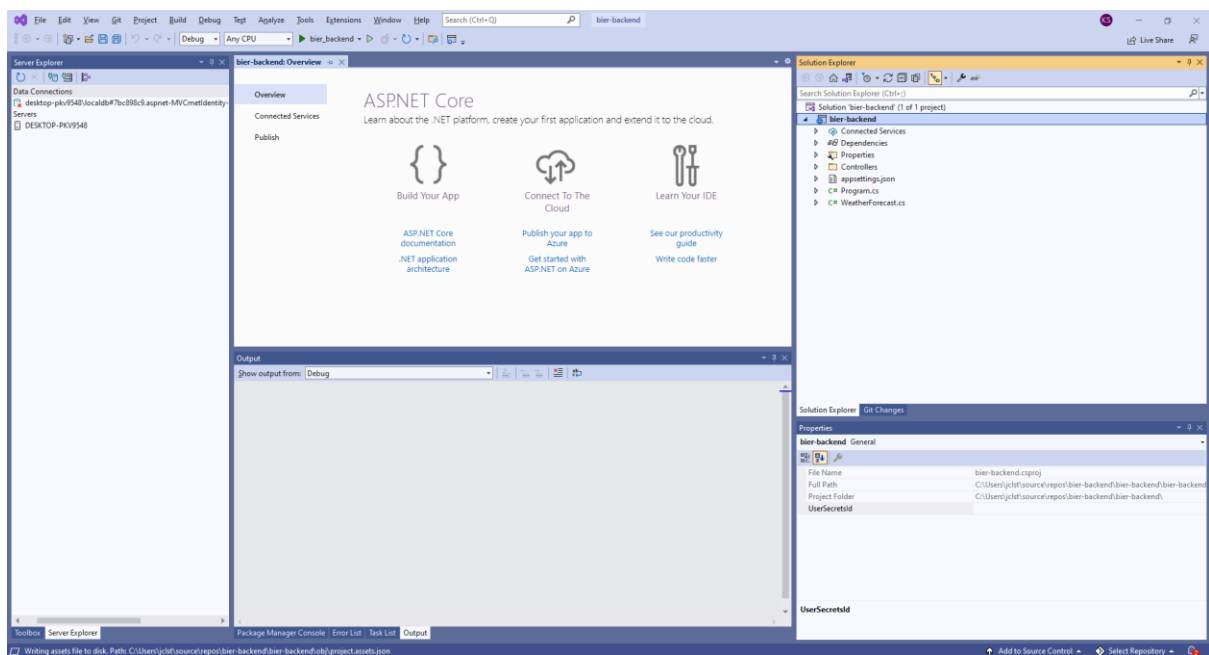
☒ Configure for HTTPS ⓘ
☐ Enable Docker ⓘ

Docker OS ⓘ
[Linux]

☒ Use controllers (uncheck to use minimal APIs) ⓘ
☒ Enable OpenAPI support ⓘ

Back Create

Je hebt nu een nieuw project , dat ziet er zo uit:



```

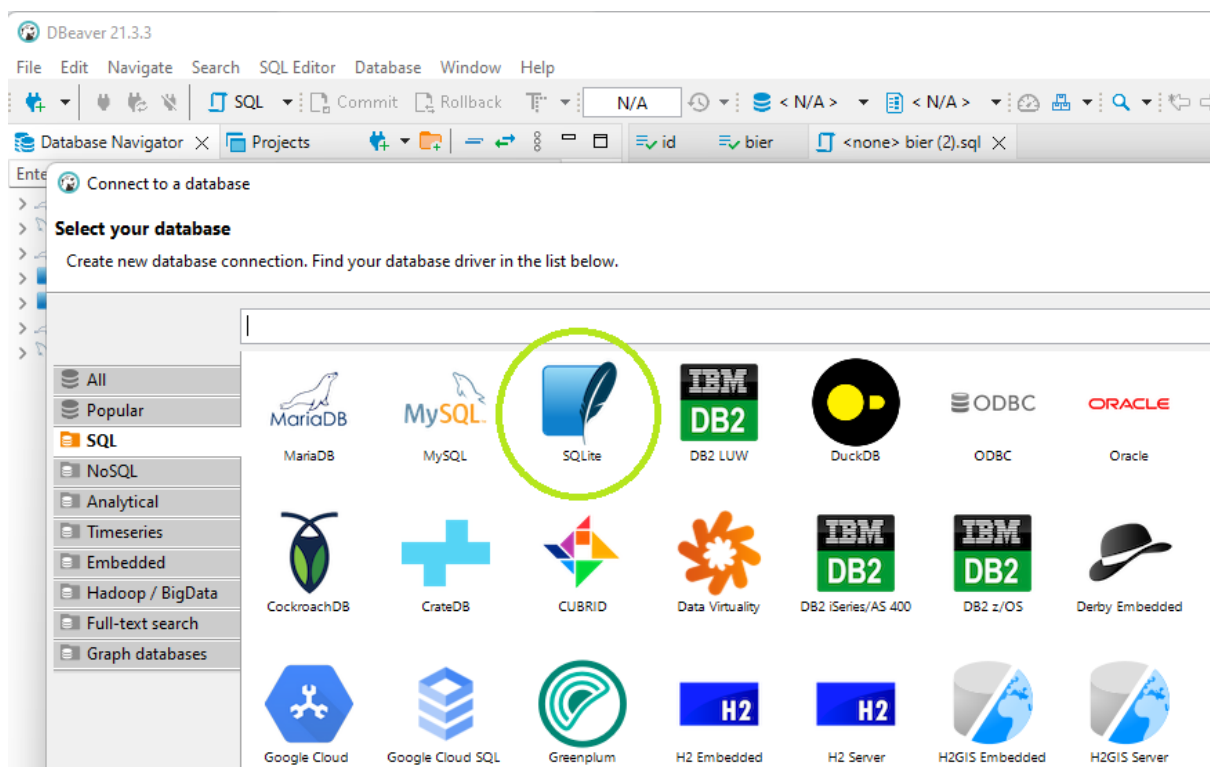
    },
    "AllowedHosts": "*",
    "ConnectionStrings": {
      "DefaultConnection": "Data Source=./bier.db"
    }
  }
}

```

Stap 4. Voeg een Connectionstring toe naar een SQLite bestand in de appsettings.json file.

Dit bestand gaan we nu aanmaken:

Stap 5. Gebruik DBeaver om een SQLite database aan te maken met de naam *bier.db* , run het sql bestand wat je krijgt van de docent of hier kunt vinden.



NB.: Gebruik een tool als dbeaver creeër hiermee een nieuwe SQLite connectie met de naam *bier.db* in je project map, en run dan het sql script om de database tabel te maken en te vullen:

```
-- bier definition
CREATE TABLE `bier` (
  `id` integer primary key AUTOINCREMENT,
  `naam` varchar(40) DEFAULT NULL,
  `brouwer` varchar(40) DEFAULT NULL,
  `type` varchar(40) DEFAULT NULL,
  `gisting` varchar(40) DEFAULT NULL,
  `perc` float DEFAULT NULL,
  `inkoop_prijs` decimal(5,2) NOT NULL
);

INSERT INTO bier (naam,brouwer,"type"
  ('Abondance mirabellen','Brouwer:
  ('Abondance pÃche','Brouwerij d
```

SQL Script.

Stap 6. Installeer de Tools voor het entity framework , en voor SQLite dit doe je in de console (Package Manager Console, of in een andere terminal console)

```
Install-Package Microsoft.EntityFrameworkCore.Tools
```

```
Install-Package Microsoft.EntityFrameworkCore.Sqlite
```

Stap 7. Nu komt de belangrijkste stap: *Voer de Scaffolding uit!!!!* Scaffolding maakt je model class en je dbcontext class, de classes die ervoor zorgen dat je kunt connecten met de database en dat je gegevens kunt inlezen. Type in de Console:

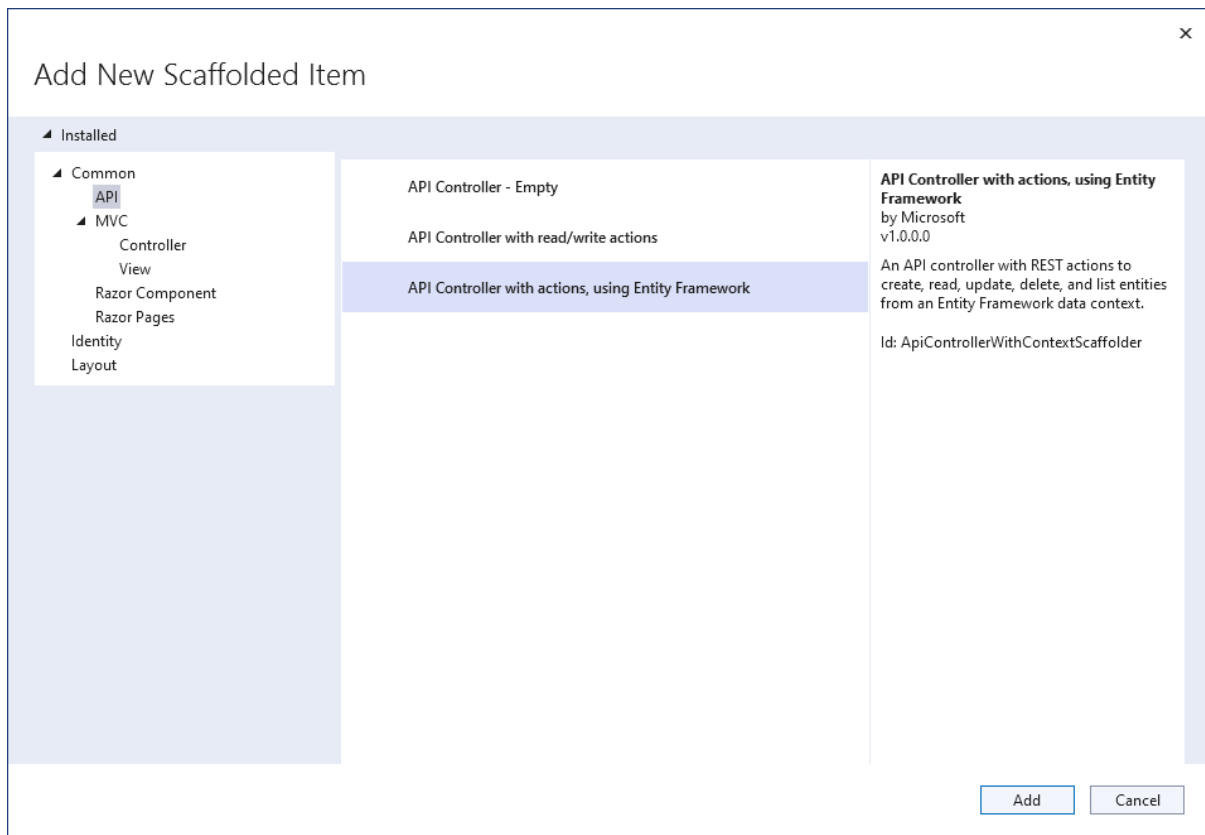
```
Scaffold-DbContext "DataSource=.\bier.db" Microsoft.EntityFrameworkCore.Sqlite
-OutputDir Models
```

Je krijgt nu een Models map erbij in je project met daarin de class bier en de class bierContext!

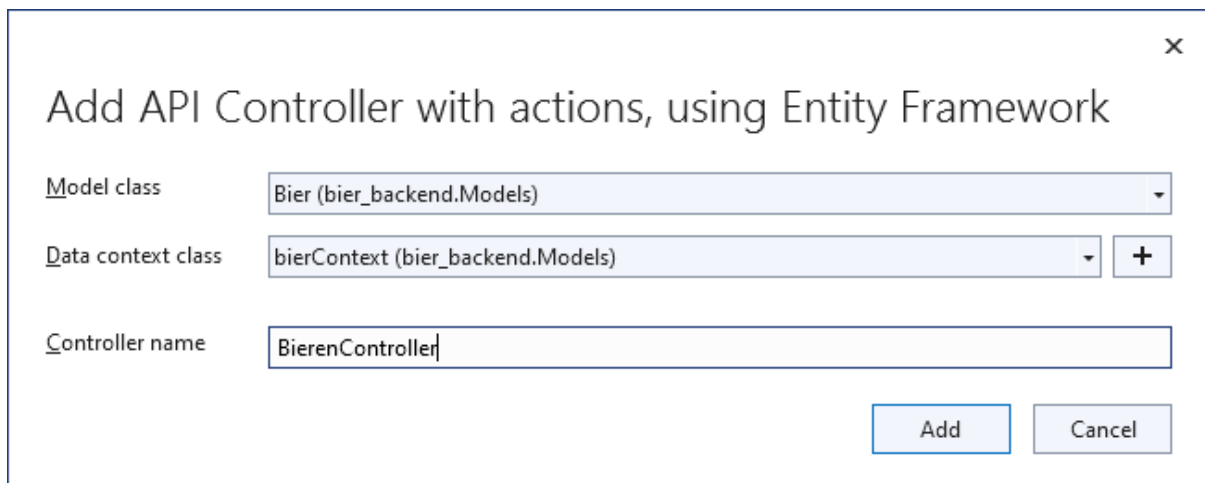
We gaan nu een Api controller maken voor bier!

Stap 8. Rechtermuisklik op de *Controllers* map en kies *Add* en dan *Controller*

Kies nu een API controller, met Entity Framework



Vul de volgende namen in, let op maak van BiersController =>BierenController dat klinkt wat beter.



Je BierenController REST API Crud is nu gemaakt.

Stap 9.

We moeten nog 1 ding doen. In je Program.cs moet je doorgeven dat je gebruik wenst te maken van de nieuw aangemaakte databaseContext met de opgegeven connectionString. Zie regel 4 en 5 in het plaatje.

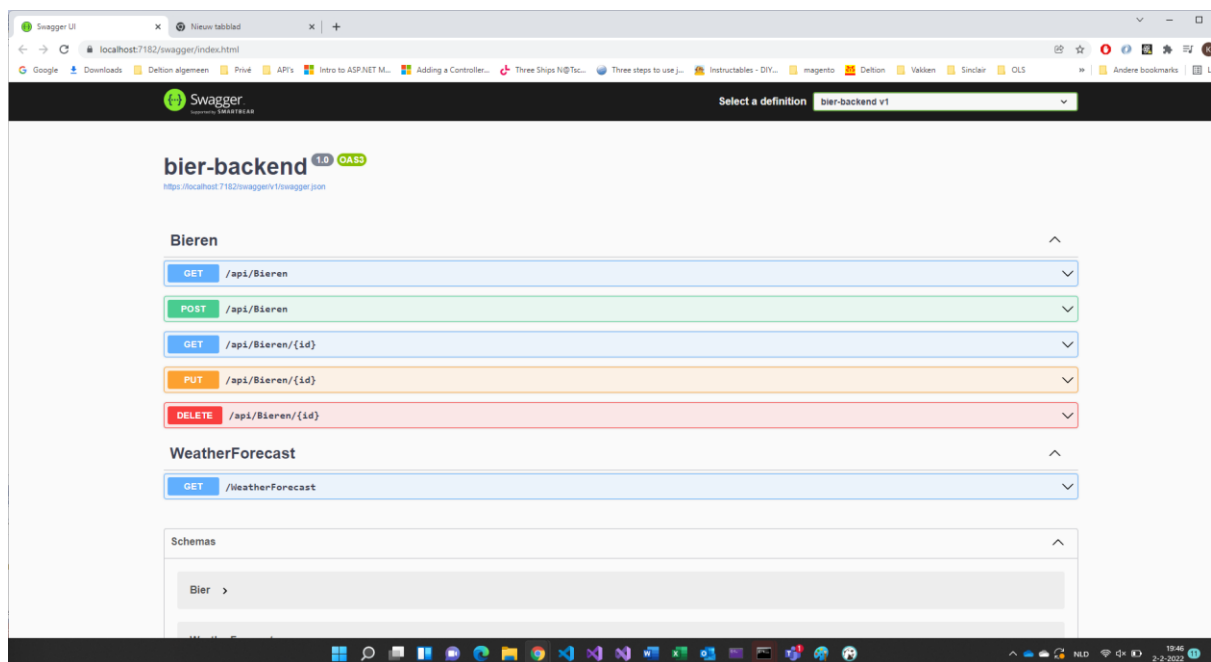
```
BierenController.cs Bier.cs bierContext.cs appsettings.json Program.cs* x bier-backend: Overview
bier-backend
1 var builder = WebApplication.CreateBuilder(args);
2
3 // Add services to the container.
4 string connectionString = builder.Configuration.GetConnectionString("DefaultConnection");
5 builder.Services.AddDbContext<bierContext>(options => options.UseSqlite(connectionString));
6
7 builder.Services.AddControllers();
8 // Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
9 builder.Services.AddEndpointsApiExplorer();
10 builder.Services.AddSwaggerGen();
11
12 var app = builder.Build();
```

Stap 10 . Right-click op de rood onderstreepte namen en resolve ze (Quick Actions & Refactoring)

Resolven wil zeggen dat je de correcte namespaces ook gebruikt in dit bestand, de volgende regels worden dan bovenin Program.cs toegevoegd:

```
using bier_backend.Models;
using Microsoft.EntityFrameworkCore;
```

Stap 11. Als je op de Start knop duwt of op F5 start je backend project op, je komt nu automatisch in de SWAGGER omgeving, waar je je API kunt testen, je ziet hier ook een voorbeeld Get controller WeatherForecast deze wordt standaard als voorbeeld meegeleverd en kun je uit de code verwijderen als je deze niet wilt:



Je kunt nu je api testen! Succes!!!!

FINISH