Creating a Database in .Net/C# Web API

DS3103 Webutvikling

Rolando Gonzalez, 2022

About this slideseries

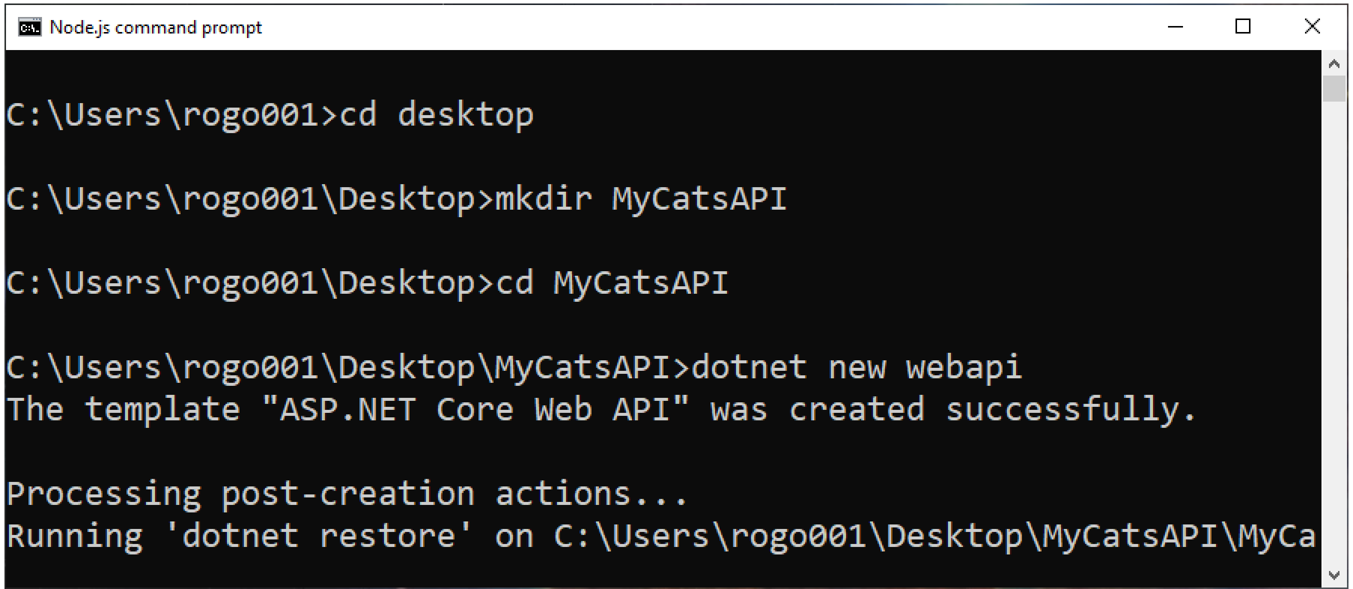
- This slideseries takes you through how to create a Database from Model classes in a Web API project, step by step from creating the Web API project.

Content

- Creating the Web API project
- Adding the Database package
- Creating the Model class(es)
- Creating the Database Context class
- Setting up Program.cs to use Database
- Migrating and creating the database
- Setting up context for CRUD in Controller

Creating the Web API project

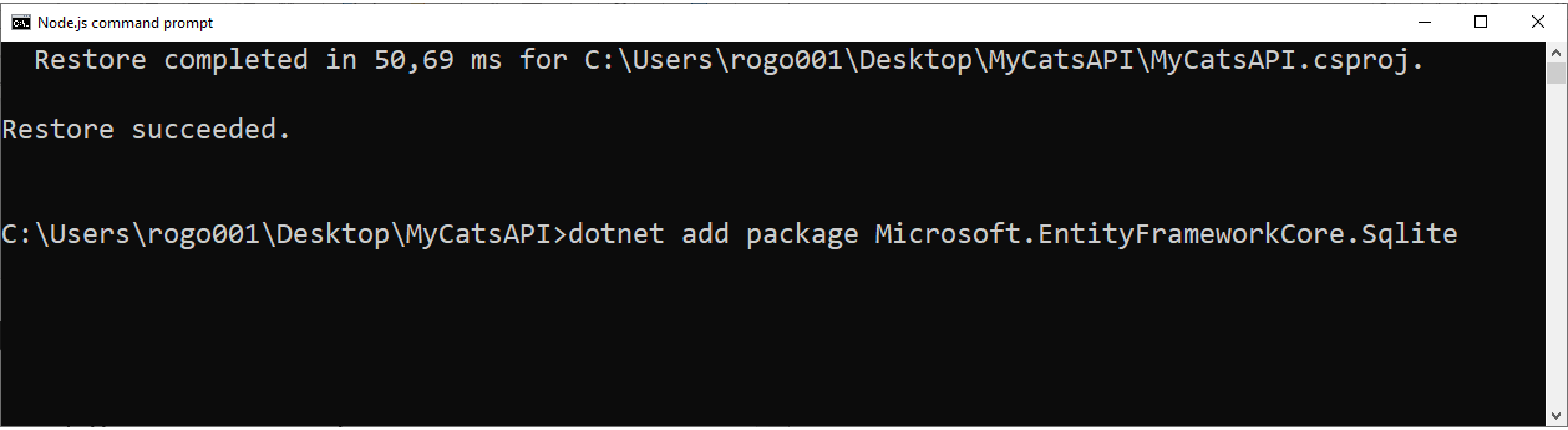
- Open the Terminal (Mac) or CMD (Ledetekst, Windows).
- Create a new folder in a location you want to and run the command for creating a new Web API project. **dotnet new webapi**



```
C:\Users\rogo001>cd desktop

C:\Users\rogo001\Desktop>mkdir MyCatsAPI

C:\Users\rogo001\Desktop>cd MyCatsAPI

C:\Users\rogo001\Desktop\MyCatsAPI>dotnet new webapi
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on C:\Users\rogo001\Desktop\MyCatsAPI\MyCa
```

Add package for Database

- **Add the package for Sqlite:**
  - **dotnet add package Microsoft.EntityFrameworkCore.Sqlite**



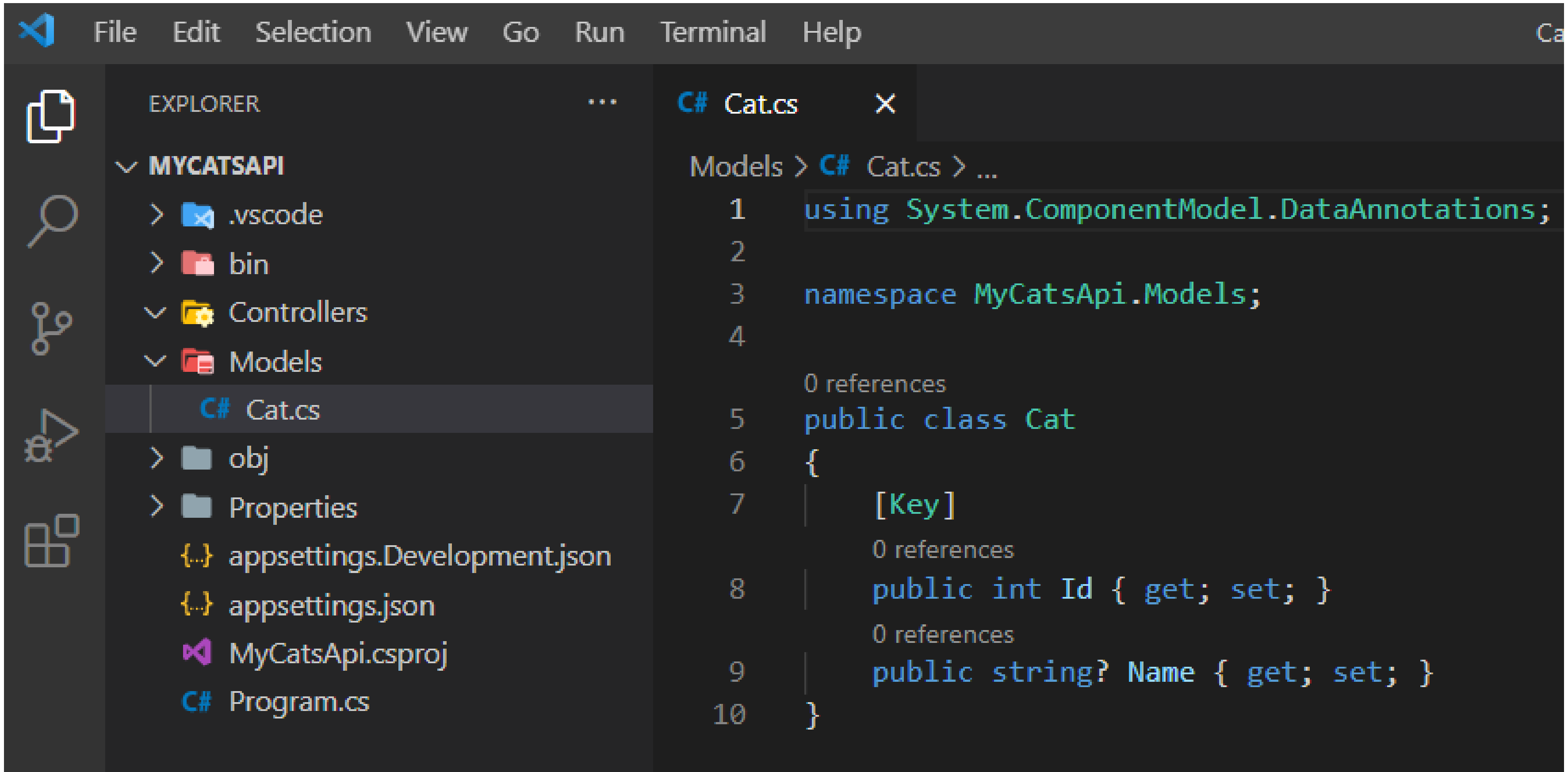
```
  Restore completed in 50,69 ms for C:\Users\rogo001\Desktop\MyCatsAPI\MyCatsAPI.csproj.

Restore succeeded.


C:\Users\rogo001\Desktop\MyCatsAPI>dotnet add package Microsoft.EntityFrameworkCore.Sqlite
```

Add dotnet EF

- PS! The following command should be enough to do only once on your machine!
- Run command to install Entity Framework (to make queries against the Database)
  - **dotnet tool install -g dotnet-ef**

Add the Model class

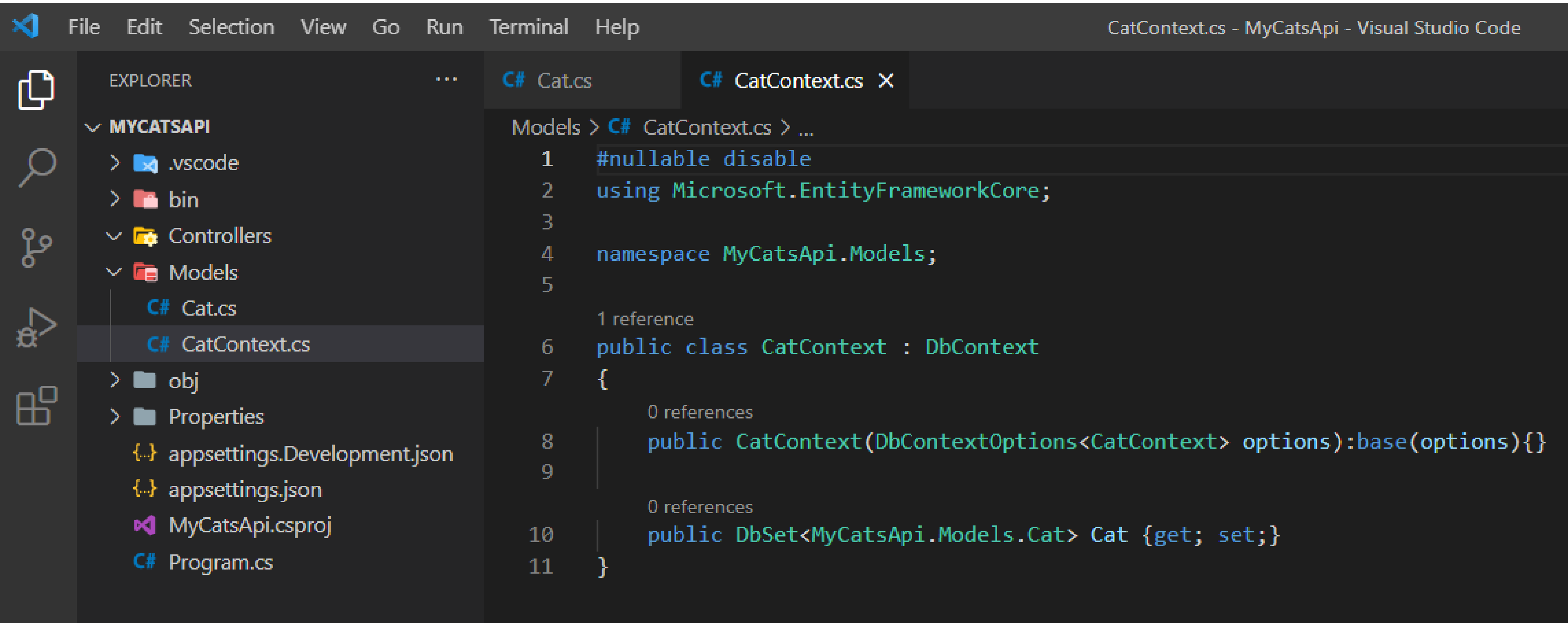
- Add a Models folder and a Model class Cat
- The Id is necessary for the Database



```csharp
using System.ComponentModel.DataAnnotations;

namespace MyCatsApi.Models;

0 references
public class Cat
{
    [Key]
    0 references
    public int Id { get; set; }
    0 references
    public string? Name { get; set; }
}
```

Add the database context class

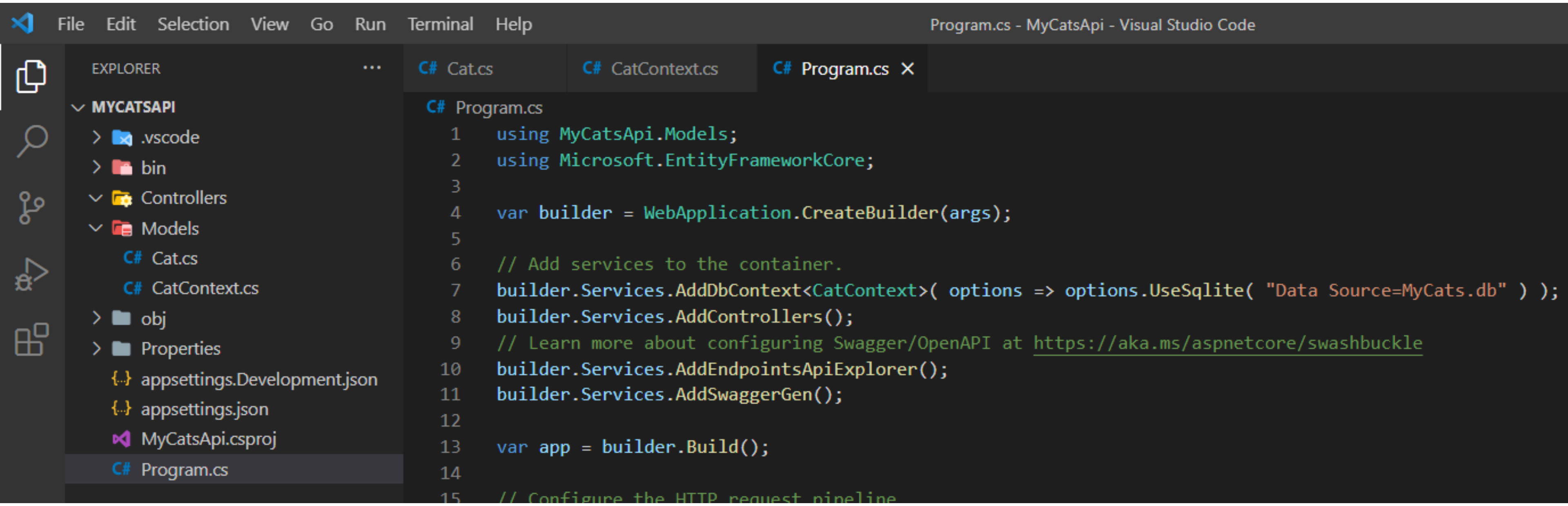
- Inside the Models folder you will add the class which defines which Model classes to include in the DB



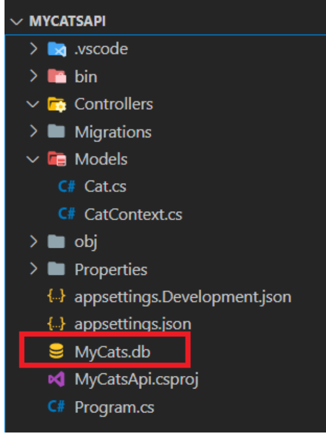
```
#nullable disable
using Microsoft.EntityFrameworkCore;

namespace MyCatsApi.Models;

1 reference
public class CatContext : DbContext
{
    0 references
    public CatContext(DbContextOptions<CatContext> options):base(options){}

    0 references
    public DbSet<MyCatsApi.Models.Cat> Cat {get; set;}
}
```

Program.cs: register Db

- include using statements and add services.AddDbContext. Here you set the name of the database (MyCats.db here).



```
using MyCatsApi.Models;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.
builder.Services.AddDbContext<CatContext>( options => options.UseSqlite( "Data Source=MyCats.db" ) );
builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline
```

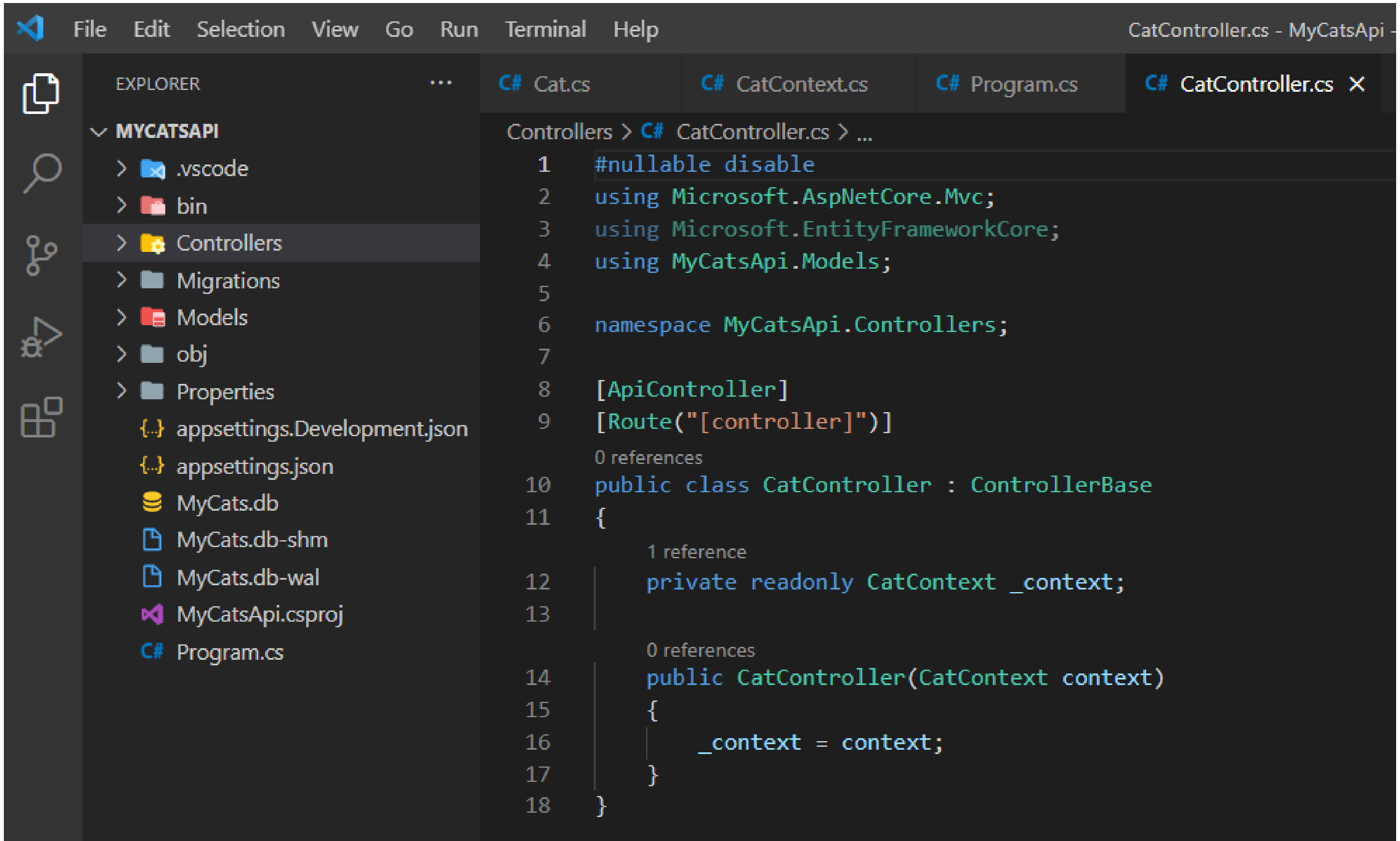


Migrating and creating DB

- You will need to run three commands to create the database in Terminal /CMD:
    1. dotnet add package Microsoft.EntityFrameworkCore.Design
    2. dotnet ef migrations add InitialCreate
    3. dotnet ef database update
- The Database file should now be visible in your project folder.

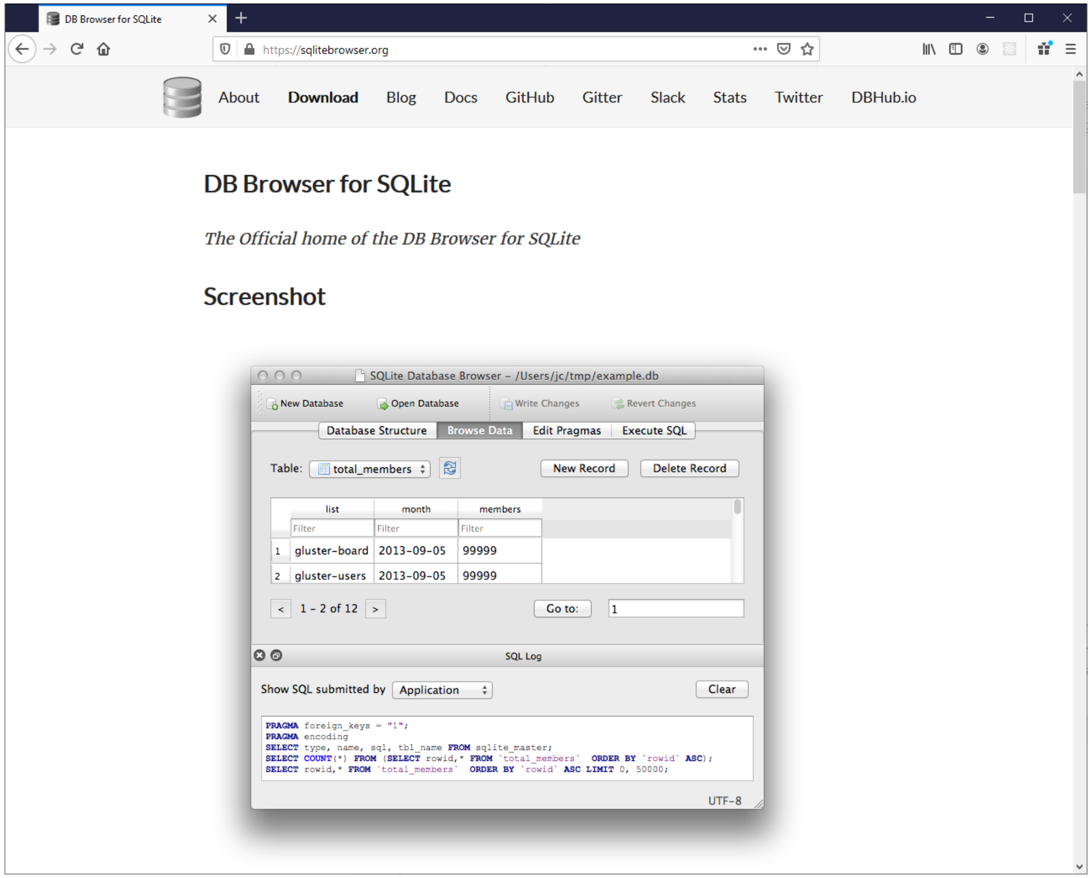
Setting up context in Controller

- To be able to use CRUD against the DB, you setup the context class in your controller.



```
#nullable disable
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MyCatsApi.Models;

namespace MyCatsApi.Controllers;

[ApiController]
[Route("[controller]")]
0 references
public class CatController : ControllerBase
{
    1 reference
    private readonly CatContext _context;

    0 references
    public CatController(CatContext context)
    {
        _context = context;
    }
}
```

DB Browser



DB Browser for SQLite

The Official home of the DB Browser for SQLite

Screenshot

Table: total_members

	list	month	members
	Filter	Filter	Filter
1	gluster-board	2013-09-05	99999
2	gluster-users	2013-09-05	99999

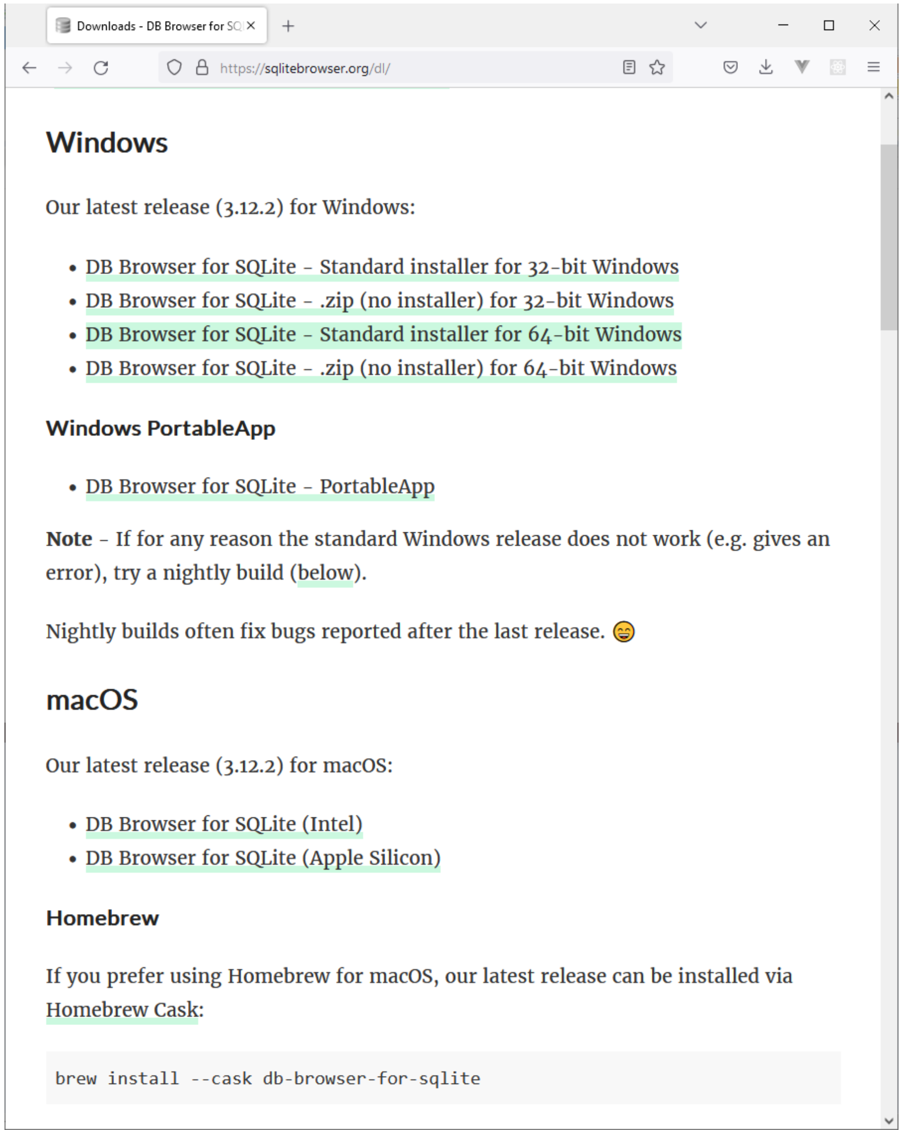
< 1 - 2 of 12 >   Go to: 1

Show SQL submitted by: Application

```
PRAGMA foreign_keys = "1";
PRAGMA encoding
SELECT type, name, sql, tbl_name FROM sqlite_master;
SELECT COUNT(*) FROM (SELECT rowid,* FROM `total_members`  ORDER BY `rowid` ASC);
SELECT rowid,* FROM `total_members`  ORDER BY `rowid` ASC LIMIT 0, 50000;
```

Checking database in DB Browser

- DB Browser is a tool for seeing metainformation and information about a database. You can also edit the information inside it.
- Download DB Browser for your os: https://sqlitebrowser.org/dl/



Windows

Our latest release (3.12.2) for Windows:

- DB Browser for SQLite – Standard installer for 32-bit Windows
- DB Browser for SQLite – .zip (no installer) for 32-bit Windows
- DB Browser for SQLite – Standard installer for 64-bit Windows
- DB Browser for SQLite – .zip (no installer) for 64-bit Windows

Windows PortableApp

- DB Browser for SQLite – PortableApp

Note – If for any reason the standard Windows release does not work (e.g. gives an error), try a nightly build (below).

Nightly builds often fix bugs reported after the last release. 😄

macOS

Our latest release (3.12.2) for macOS:

- DB Browser for SQLite (Intel)
- DB Browser for SQLite (Apple Silicon)

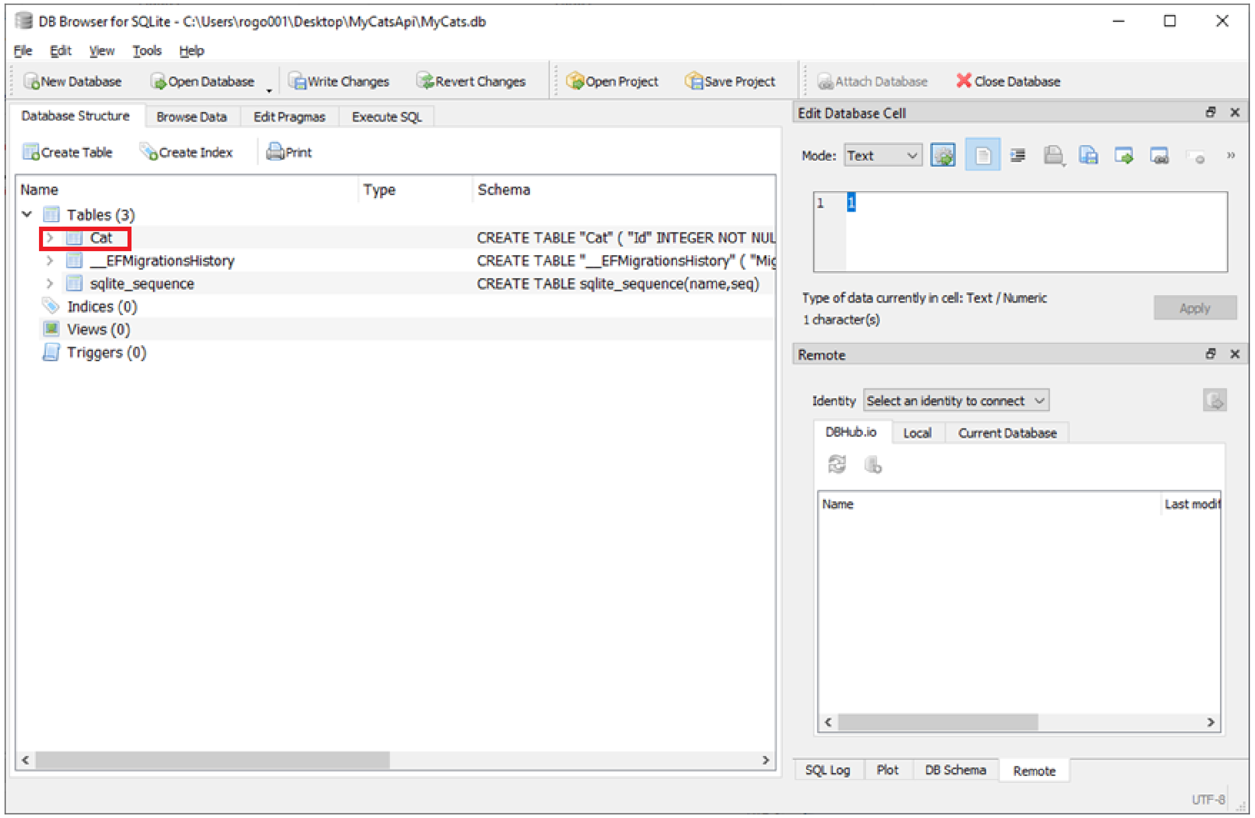
Homebrew

If you prefer using Homebrew for macOS, our latest release can be installed via Homebrew Cask:

```
brew install --cask db-browser-for-sqlite
```

MyCats.db in DB Browser

- Choose Open Database and find your database
- Click on the Cat table
- Choose the Browse Data tab



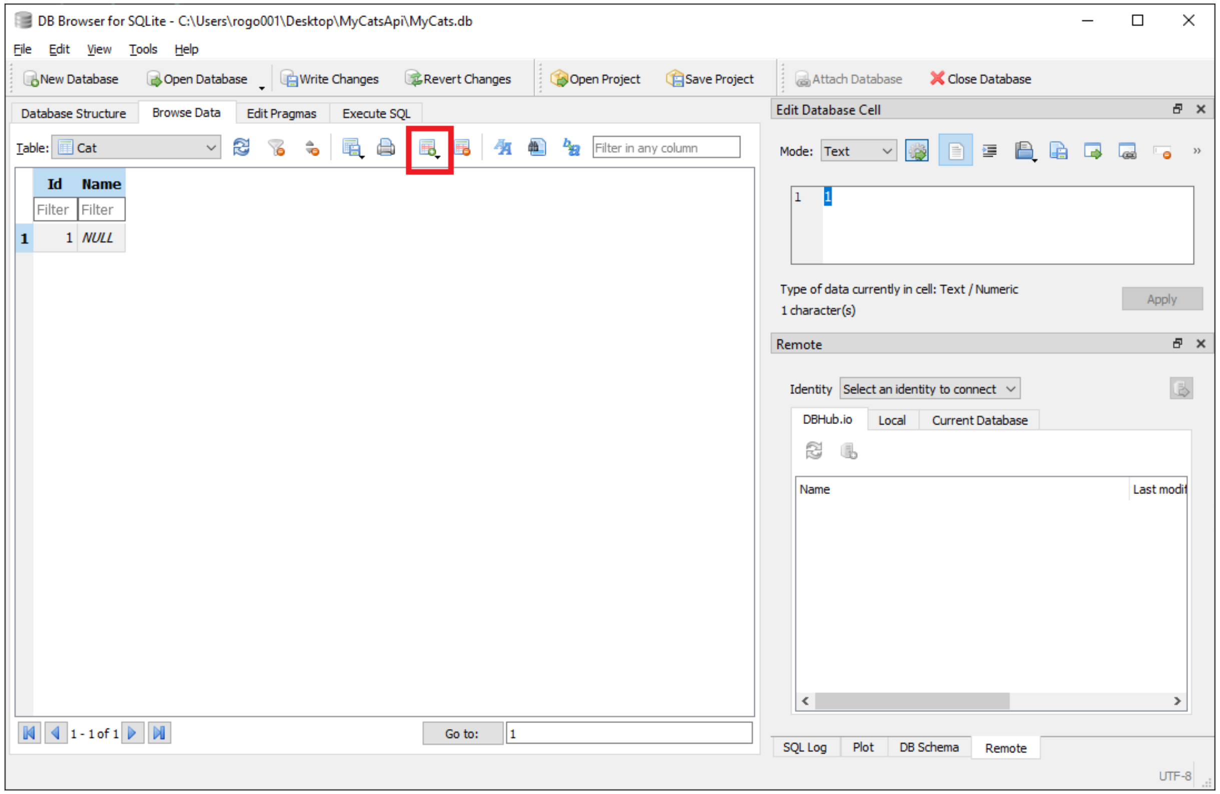
Name	Type	Schema
Tables (3)		
Cat		CREATE TABLE "Cat" ("Id" INTEGER NOT NUL
__EFMigrationsHistory		CREATE TABLE "__EFMigrationsHistory" ("Mig
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Indices (0)		
Views (0)		
Triggers (0)		

```
1  1
```

Type of data currently in cell: Text / Numeric
1 character(s)

Adding a record (entitet) in the db

- Click on the New Record button to add a new records in the database. Click Write Changes when done. You may also want to close the program.



	Id	Name
	Filter	Filter
1	1	NULL

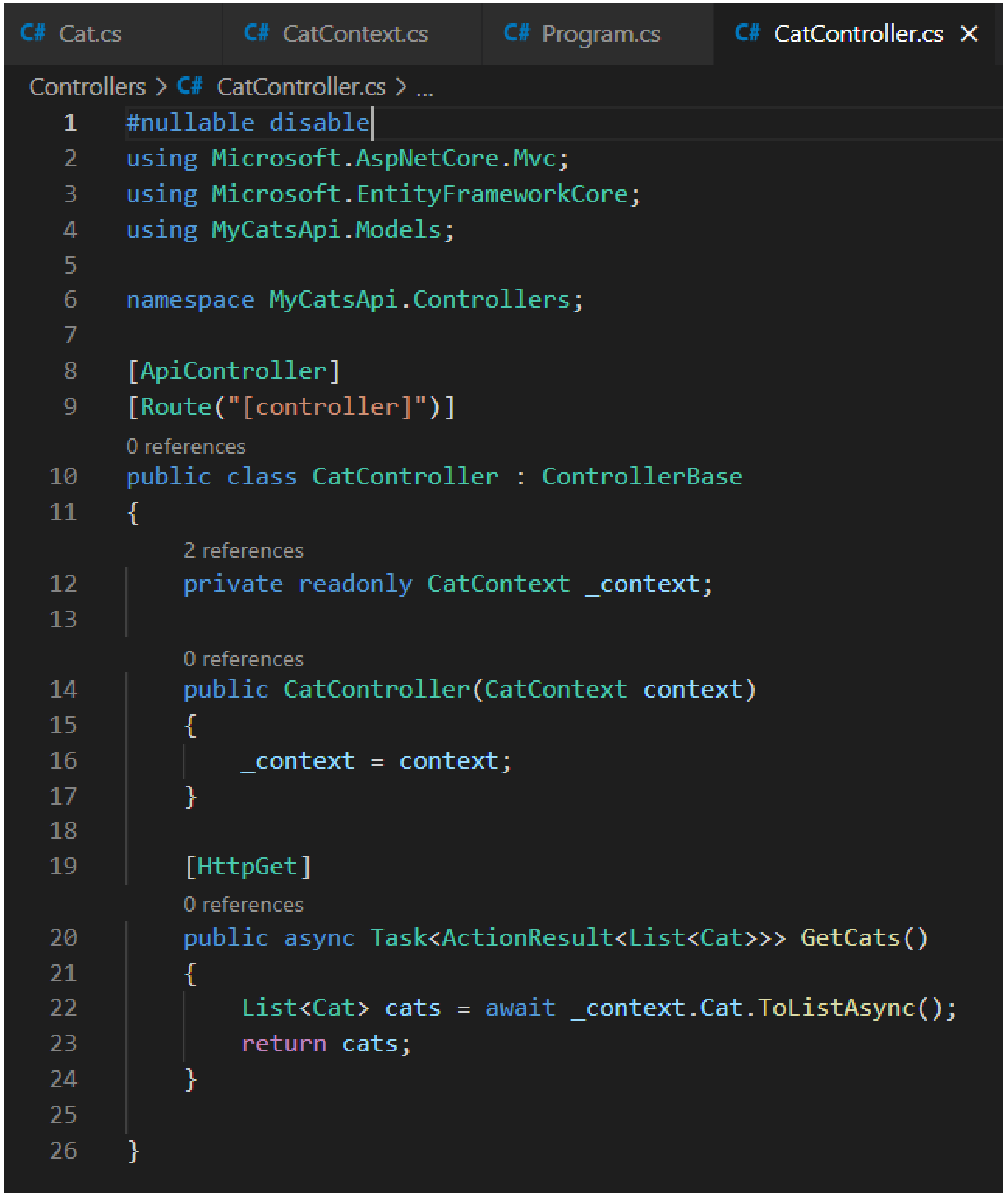
Type of data currently in cell: Text / Numeric
1 character(s)

c**R**UD

Read - GET

Read / GET test

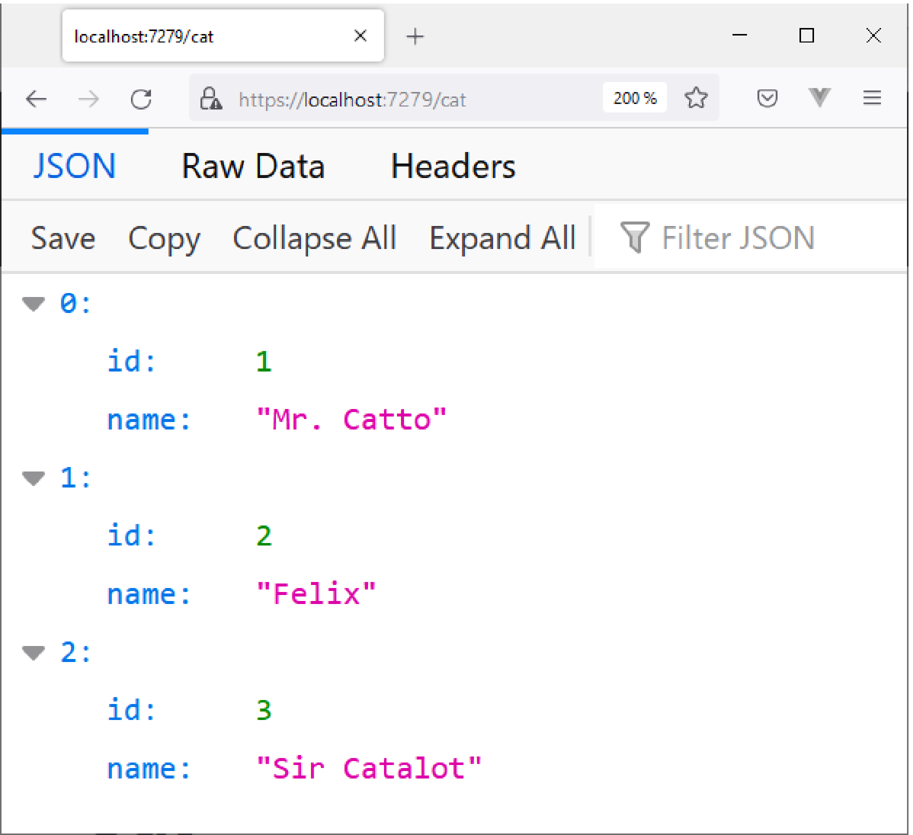
- After adding cats with DB Browser, make a HttpGet method to get all cats



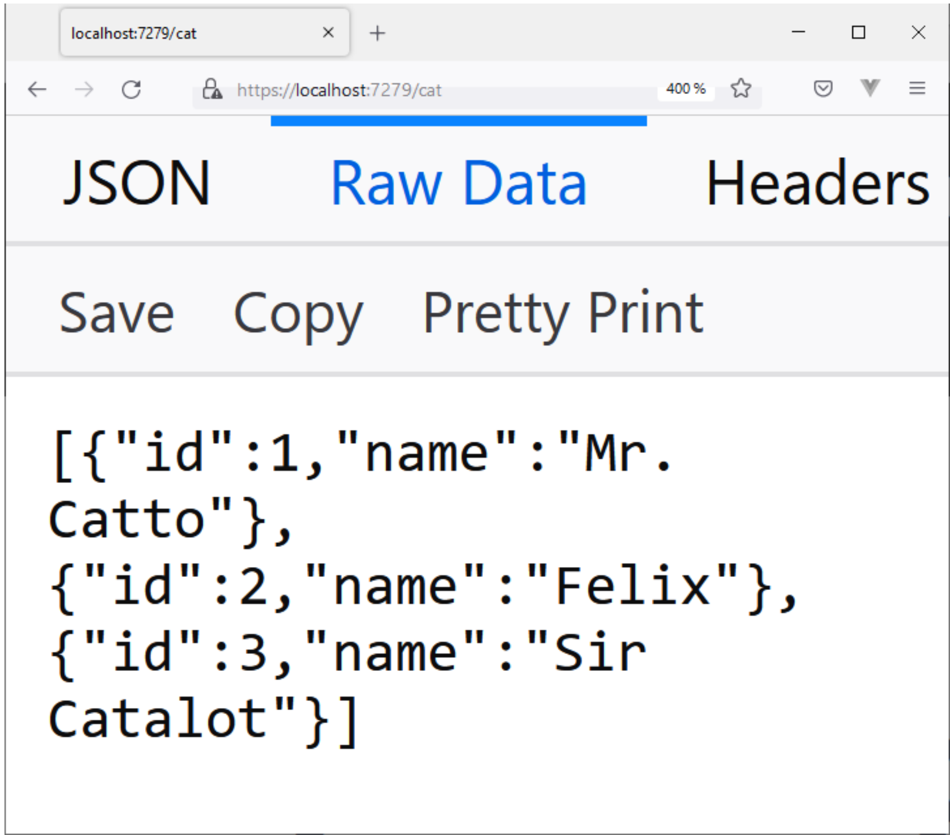
```
#nullable disable
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MyCatsApi.Models;

namespace MyCatsApi.Controllers;

[ApiController]
[Route("[controller]")]
0 references
public class CatController : ControllerBase
{
    2 references
    private readonly CatContext _context;

    0 references
    public CatController(CatContext context)
    {
        _context = context;
    }

    [HttpGet]
    0 references
    public async Task<ActionResult<List<Cat>>> GetCats()
    {
        List<Cat> cats = await _context.Cat.ToListAsync();
        return cats;
    }

}
```

Read test

- Running the Get should give you a result with all saved Cat objects



```
0:
    id:      1
    name:    "Mr. Catto"
1:
    id:      2
    name:    "Felix"
2:
    id:      3
    name:    "Sir Catalot"
```



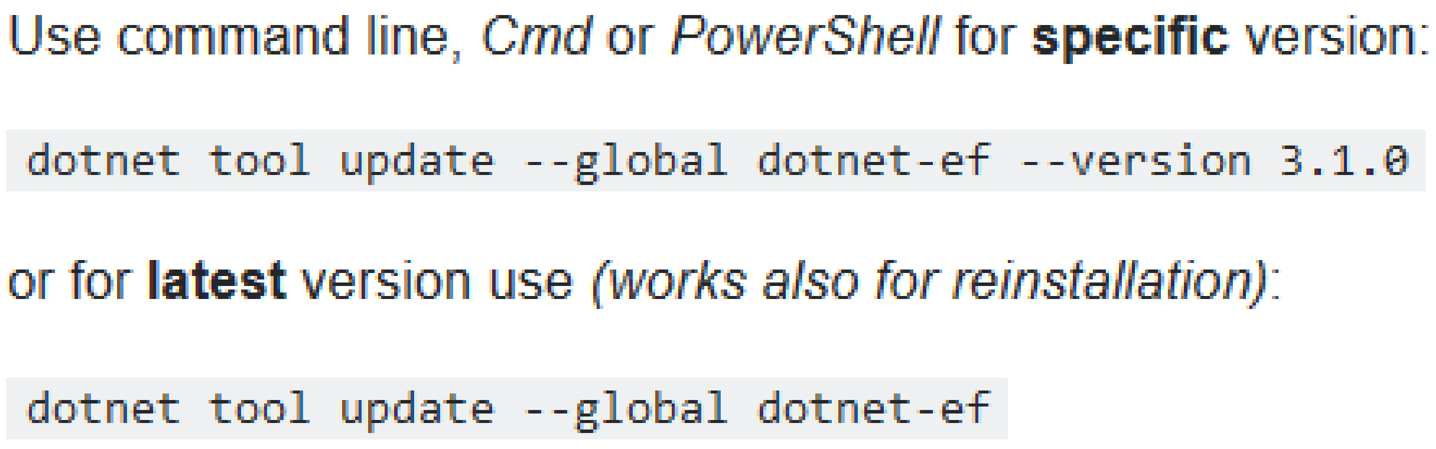
```
[{"id":1,"name":"Mr.
Catto"},
{"id":2,"name":"Felix"},
{"id":3,"name":"Sir
Catalot"}]
```

Notice about Id

- Notice that the Id is autogenerated by the Database.

Need to update EF?

- If you get information about that your EF version is old, you can update to newer/newest version.



Use command line, *Cmd* or *PowerShell* for **specific** version:

```
dotnet tool update --global dotnet-ef --version 3.1.0
```

or for **latest** version use *(works also for reinstallation)*:

```
dotnet tool update --global dotnet-ef
```

Mac «*Fatal error» with dotnet ef*?

- https://stackoverflow.com/questions/61568345/install-dotnet-ef-success-but-when-call-it-hit-error

More about the context class

- https://www.entityframeworktutorial.net/basics/context-class-in-entity-framework.aspx