VIJAY M

Day 18

Date-01/09/2025

create a small web api project that fetches the list of pet (dog varieties) from sql server via the code first approach entity framework approach

-- NOTE this should be completely done in the vs code

commands:

Create new Web API project

dotnet new webapi -n MyApiProject

Install EF Core packages

For SQL Server and tools:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Tools

Add migration

dotnet ef migrations add viavscode

Update database

dotnet ef database update

Install Swagger (Swashbuckle)

dotnet add package Swashbuckle.AspNetCore

Build project

dotnet build

Run project

dotnet run

Source Code:

program.cs

```
using Microsoft.EntityFrameworkCore;
using MyApiProject.Context;
using Microsoft.OpenApi.Models;
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddControllers();
                    const string connectionstring = "Data Source=PTPLL605;" +
                "Initial Catalog=sampledb;" +
                "Integrated Security=True;" +
          "TrustServerCertificate=True;";
  builder.Services.AddDbContext<MyAppDbContext>(options =>
                            options.UseSqlServer(connectionstring));
builder.Services.AddEndpointsApiExplorer(); // Enables endpoint exposure
builder.Services.AddSwaggerGen();                    // Adds Swagger generation
var app = builder.Build();
if (app.Environment.IsDevelopment())
{
      app.UseSwagger();
      app.UseSwaggerUI(); // Default route: /swagger
}
app.UseRouting();
app.MapControllers();
app.Run();
```

Model/Dog.cs

```
namespace MyApiProject.Model{
```

```
    public class Dog{
        public int Id { get; set; }
        public string Breed { get; set; } = string.Empty;
        public string Size { get; set; } = string.Empty; // e.g., Small, Medium, Large
        public string Origin { get; set; } = string.Empty;
        public bool IsHypoallergenic { get; set; }
    }
}
```

Controllers/DogController.cs

```
using MyApiProject.Model;
using MyApiProject.Context;
using Microsoft.AspNetCore.Mvc;
namespace MyApiProject.Controllers{
[ApiController]
[Route("[controller]")]
public class DogsController : ControllerBase
{
    private MyAppDbContext context {get;set;}
    public DogsController(MyAppDbContext context){
        this.context=context;
    }
    [HttpGet("getall")]
    public ActionResult<IEnumerable<Dog>> GetAll()
    {
        var dogs = context.dog.ToList();
```

```
            return Ok(dogs);
        }
          [HttpPost("adddog")]
        public IActionResult AddDog([FromBody] Dog newDog)
        {
            context.dog.Add(newDog);
            context.SaveChanges();
            return Ok("added successfully");
        }
}
}
```

Context/MyAppDbContext.cs

```
using MyApiProject.Model;
using Microsoft.EntityFrameworkCore;
namespace MyApiProject.Context{
    public class MyAppDbContext:DbContext{
        public MyAppDbContext(DbContextOptions<MyAppDbContext> options): base(options){}
        public DbSet<Dog> dog {get;set;}
    }
}
```

Output

Dogs/getall

Request URL

```
http://localhost:5287/Dogs/getall
```

Server response

Code	Details
200	Response body

```
[
  {
    "id": 1,
    "breed": "Labrador",
    "size": "Large",
    "origin": "Canada",
    "isHypoallergenic": false
  },
  {
    "id": 2,
    "breed": "Shiba Inu",
    "size": "Medium",
    "origin": "Japan",
    "isHypoallergenic": false
  }
]
```

Response headers

```
content-type: application/json; charset=utf-8
date: Mon,01 Sep 2025 11:44:31 GMT
server: Kestrel
transfer-encoding: chunked
```

Dogs/adddog

Curl

```
curl -X 'POST' \
  'http://localhost:5287/Dogs/adddog' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
  "id": 0,
  "breed": "Shiba Inu",
  "size": "Medium",
  "origin": "Japan",
  "isHypoallergenic": false
}
'
```

Request URL

```
http://localhost:5287/Dogs/adddog
```

Server response

Code	Details
200	Response body

```
added successfully
```

Response headers

```
content-type: text/plain; charset=utf-8
date: Mon,01 Sep 2025 11:44:24 GMT
server: Kestrel
transfer-encoding: chunked
```