TI-Capital Humano

Desarrollador .Net

Crear Web API con .Net Core y EntityFramework

Contenido

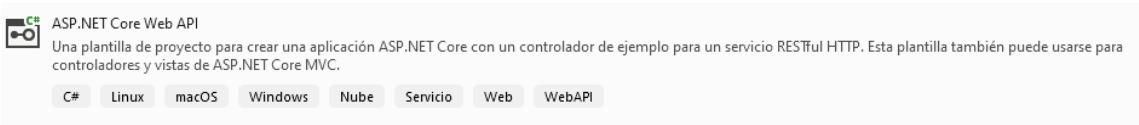
1. Crear WebAPI

Crear una Web API con acciones que usan EntityFramework, utilizando .Net Core 6., las cuales son acciones de un CRUD, utilizando únicamente las plantillas y el scarfolding.

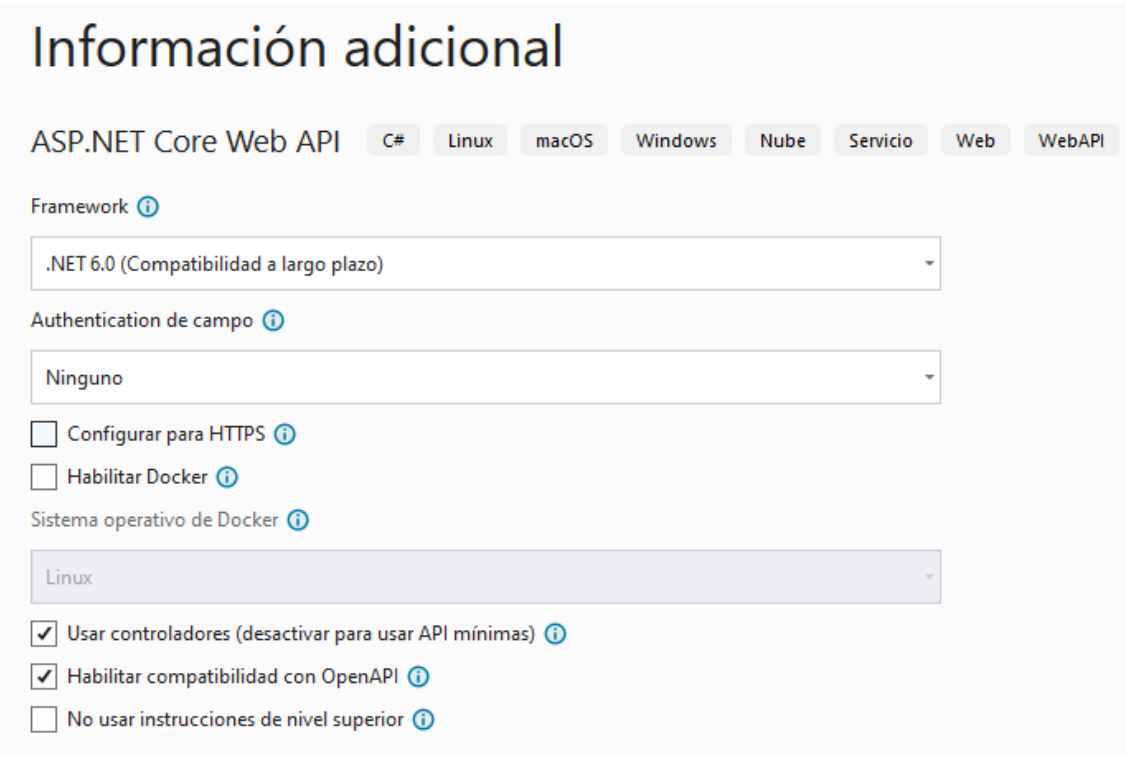
Las acciones CRUD serán para la tabla Estados de la base de datos InstitutoTich

1.1. Crear Proyecto

Crear un Proyecto tipo ASP .Net Core Web API, con el nombre de EstadosAPI



Eliminar la configuración para https



1.2. Agregar Paquetes Nugets

Como se utilizará EntityFramework agregar los siguientes paquetes:

Microsoft.EntityFrameworkCore.SqlServer ver 6.0.29
Microsoft.EntityFrameworkCore.Tools ver 6.0.29

1.3. Mapear la Base de datos con EntityFrameWork

Utilizando ingeniería inversa ejecutar el proceso de scaffolding para obtener las clases de tipo de entidad y una clase DbContext basada en un esquema de base de datos. Utilizar el comando Scaffold-DbContext de las herramientas de consola del administrador de paquetes (PMC) de EF Core

Primeramente crear la carpeta **Models** dentro del proyecto para que colocar ahí la clase de contexto y la de entidad resultante. En esta caso solo se tomará la Tabla Estados de la Base de Datos InstitutoTich

Scaffold-DbContext "Server=MSI;Database=InstitutoTich;User ID=sa;password=Pass2017;" Microsoft.EntityFrameworkCore.SqlServer -Tables "Estados" -Context EdoContext -ContextDir Models -OutputDir Models -NoPluralize -Force

Toda vez que se há creado la clase de DBContext y la de Entidad

Se deberá ajustar el método OnConfiguring de la clase de Contexto, ya que en ella el scarfolding deja las credenciales de la base de datos, mismas que deberán quedar en el archivo de configuración **appsettings.json**

```
public partial class EdoContext : DbContext
{
    0 referencias
    public EdoContext()
    {
    }

    0 referencias
    public EdoContext(DbContextOptions<EdoContext> options)
        : base(options)
    {
    }

    0 referencias
    public virtual DbSet<Estados> Estados { get; set; } = null!;

    0 referencias
    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        if (!optionsBuilder.IsConfigured)
        {
arning To protect potentially sensitive information in your connection string, you should move it out of s
            optionsBuilder.UseSqlServer("Server=MSI;Database=InstitutoTich;User ID=sa;password=Pass2017;
        }
    }
```

La cadena de conexión se pasará al archivo de configuración

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
  "ConnectionStrings": {
    "InstitutoTich": "Server=MSI;Database=InstitutoTich;User ID=sa;password=Pass2017;"
  }
}
```

Además en el archivo **program.cs** se deberá cargar esta cadena de conexión como sigue:

```
using Microsoft.EntityFrameworkCore;
using EstadosAPI.Models;
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();
// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

builder.Services.AddDbContext<EdoContext>(opcion => {
    opcion.UseSqlServer(builder.Configuration.GetConnectionString("InstitutoTich"));
});
var app = builder.Build();
```

Agregar las referencias para la clase UseSqlServer y para la clase de contexto

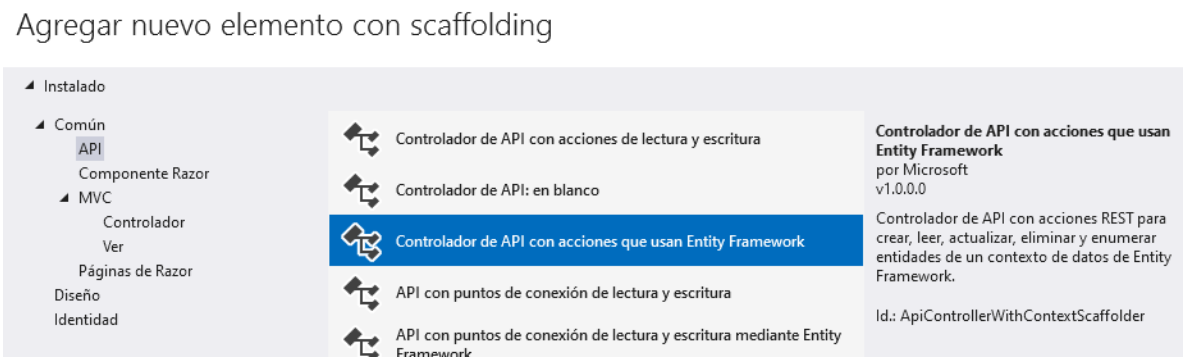
El builder.Configuration obtendrá la cadena de conexión del archivo de configuración que se encuentre con el nombre proporcionado, en este caso InstitutoTich, y el builder.Services.AddDbContext se la pasa a la clase de contexto.

Hecho lo anterior, se deberá ajustar el método OnConfiguring de la clase de Contexto, dejando el cuerpo vacio.

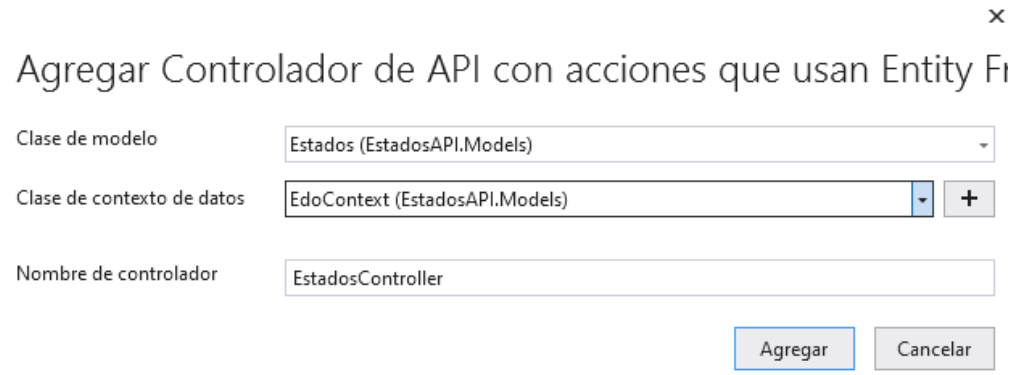
```
0 referencias
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
}
```

1.4. Crear el Controlador

Crear el controlador de API con acciones que usan Entity Framework



Seleccionar la clase de modelo y la clase de contexto, en este caso Estados y EdoContext respectivamente



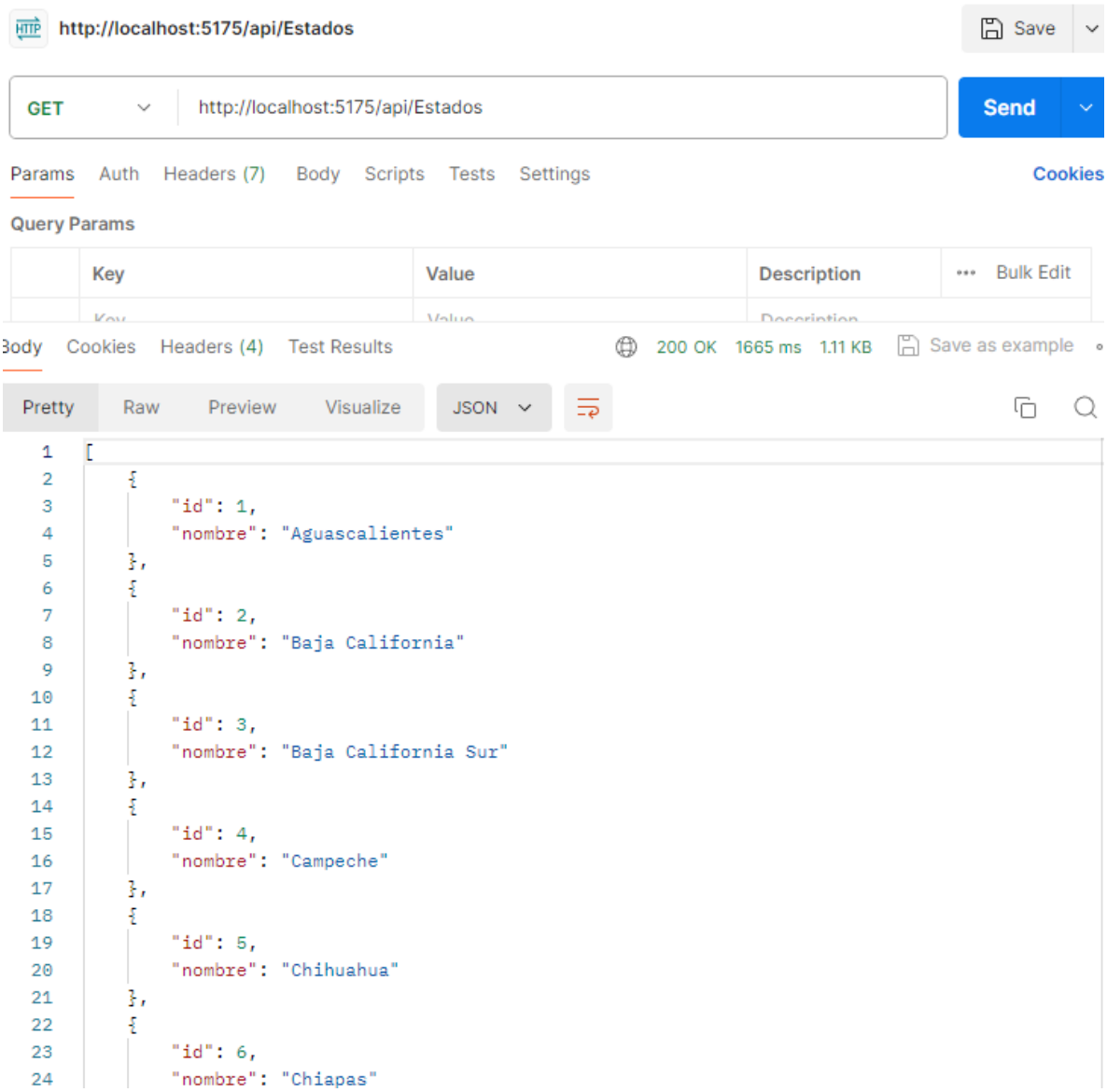
Con esto el scarfolding creará el controlador con los métodos de un CRUD para el modelo seleccionado

```
namespace EstadosAPI.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 referencia
    public class EstadosController : ControllerBase
    {
        private readonly EdoContext _context;

        0 referencias
        public EstadosController(EdoContext context)...

        // GET: api/Estados
        [HttpGet]
        0 referencias
        public async Task<ActionResult<IEnumerable<Estados>>> GetEstados()...

        // GET: api/Estados/5
        [HttpGet("{id}")]
        0 referencias
        public async Task<ActionResult<Estados>> GetEstados(int id)...

        // PUT: api/Estados/5 ...
        [HttpPut("{id}")]
        0 referencias
        public async Task<IActionResult> PutEstados(int id, Estados estados)...

        // POST: api/Estados ...
        [HttpPost]
        0 referencias
        public async Task<ActionResult<Estados>> PostEstados(Estados estados)...

        // DELETE: api/Estados/5
        [HttpDelete("{id}")]
        0 referencias
        public async Task<IActionResult> DeleteEstados(int id)...

        2 referencias
        private bool EstadosExists(int id)...
    }
}
```

1.5. Ejecutar la Web API

Con lo hecho anteriormente esta lista la Web API con acciones de CRUD, por lo que solo hay que ejecutarlo, con lo cual se obtendrá:

Ahora se deberá probar con Postman


http://localhost:5175/api/Estados
Save
GET
http://localhost:5175/api/Estados
Send
Params
Auth
Headers (7)
Body
Scripts
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Body
Cookies
Headers (4)
Test Results
200 OK
1665 ms
1.11 KB
Save as example
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"nombre": "Aguascalientes"
},
{
"id": 2,
"nombre": "Baja California"
},
{
"id": 3,
"nombre": "Baja California Sur"
},
{
"id": 4,
"nombre": "Campeche"
},
{
"id": 5,
"nombre": "Chihuahua"
},
{
"id": 6,
"nombre": "Chiapas"