VIJAY M

11-08-25

Day-5

DbFirst:

using Db first approch in entity framework and list trees in the homepage.

Command

Scaffold-DbContext "Data Source=PTPLL605;Initial Catalog=sampledb;Integrated Security=True;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -outputDir Models -ContextDir Data -Force

Program.cs

```
using GitApplication2.Data;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<SampledbContext>();

builder.Services.AddControllersWithViews();

var app = builder.Build();

if (!app.Environment.IsDevelopment())

{

    app.UseExceptionHandler("/Home/Error");

    // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

    app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

    name: "default",

    pattern: "{controller=Home}/{action=Index}/{id?}");
```

```
app.Run();
```

SampleDbContext.cs

```
using System.Collections.Generic;
using GitApplication2.Models;
using Microsoft.EntityFrameworkCore;

namespace GitApplication2.Data;

public partial class SampledbContext : DbContext
{
    public SampledbContext()
    {
    }

    public SampledbContext(DbContextOptions<SampledbContext> options)
        : base(options)
    {
    }

    public virtual DbSet<MyTree> MyTrees { get; set; }

    public virtual DbSet<Prod> Prods { get; set; }

    public virtual DbSet<Student> Students { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
#warning To protect potentially sensitive information in your connection string, you should move it out of source code.
You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see
```

https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see https://go.microsoft.com/fwlink/?LinkId=723263.

```
            => optionsBuilder.UseSqlServer("Data Source=PTPLL605;Initial Catalog=sampledb;Integrated Security=True;TrustServerCertificate=True;");


        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<MyTree>(entity =>
            {
                entity.ToTable("myTree");


                entity.Property(e => e.Description).HasMaxLength(50);

                entity.Property(e => e.Name).HasMaxLength(20);
            });


            modelBuilder.Entity<Prod>(entity =>
            {
                entity.ToTable("prod");
            });


            modelBuilder.Entity<Student>(entity =>
            {
                entity.ToTable("students");


                entity.Property(e => e.StudentId).HasColumnName("StudentID");
            });


            OnModelCreatingPartial(modelBuilder);
        }
```

```
        partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}
```

MyTree.cs

```
using System;
using System.Collections.Generic;

namespace GitApplication2.Models;

public partial class MyTree
{
        public int Id { get; set; }

        public string Name { get; set; } = null!;

        public string Description { get; set; } = null!;

        public int NoOfTrees { get; set; }
}
```

HomeController.cs

```
using GitApplication2.Data;
using GitApplication2.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;

namespace GitApplication2.Controllers
```

```
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        SampledbContext _sampledbContext;
        public HomeController(ILogger<HomeController> logger,SampledbContext _sampledbContext)
        {
            _logger = logger;
            this._sampledbContext = _sampledbContext;
        }

        public IActionResult Index()
        {
            var alltrees = _sampledbContext.MyTrees.ToList();

            return View(alltrees);
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
```

```
    }
}
```

Home/index.cshtml

```
@{
    ViewData["Title"] = "Tree Page";
}


@model List<MyTree>
<div class="text-center">
    <h1 class="display-4">Welcomet to Tree Page</h1>



@* @{
    var trees = ViewData["allTrees"] as List<MyTree>;
} *@



    <table class="table">
        <thead>
            <tr>
                <th>Name</th>
                <th>Description</th>
                <th>No of trees</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var tree in Model)
            {
```

```
            <tr>
                <td>@tree.Name</td>
                <td>@tree.Description</td>
                <td>@tree.NoOfTrees</td>

            </tr>
        }
    </tbody>
</table>
```

Welcomet to Tree Page

Name	Description	No of trees
apple	apple from jammu	8
mango tree	mango is yellow	9
coconut	it used for cooking	10