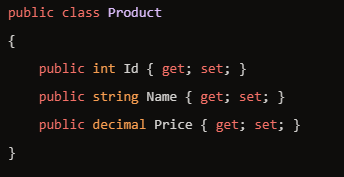
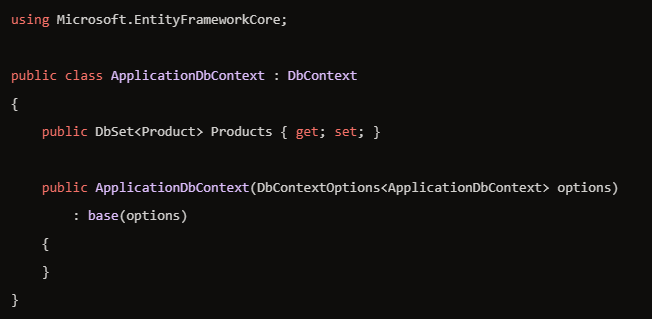
Giả sử bạn đã thiết lập thành công kết nối giữa ứng dụng ASP.NET Core MVC và cơ sở dữ liệu SQL Server. Hãy viết mã C# thực hiện các thao tác CRUD (tạo, đọc, cập nhật, xóa) cho một thực thể (entity) có tên là Product, bao gồm các thuộc tính Id, Name, và Price. Giải thích cách mã này tương tác với cơ sở dữ liệu.

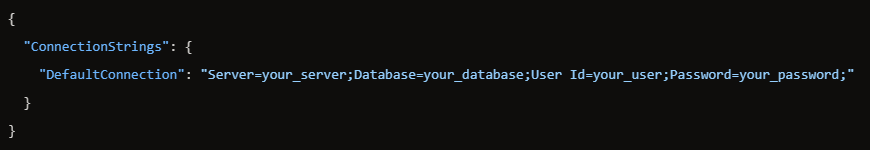
1.Trước tiên ta tạo class Product



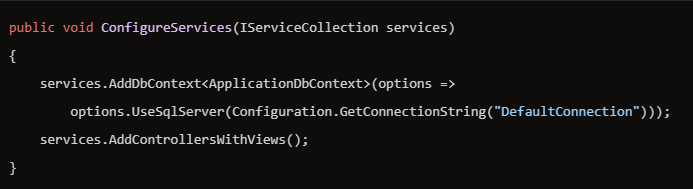
2. Tạo lớp ApplicationDbContext



3. Cấu hình kết nối cơ sở dữ liệu trong appsettings.json



4. Cấu hình dịch vụ trong Startup.cs



5. Tạo ProductsController

**using Microsoft.AspNetCore.Mvc;**

**using System.Linq;**

**using System.Threading.Tasks;**

**public class ProductsController : Controller**

**{**

**private readonly ApplicationDbContext \_context;**

**public ProductsController(ApplicationDbContext context)**

**{**

**\_context = context;**

**}**

**// GET: Products**

**public async Task<IActionResult> Index()**

**{**

**return View(await \_context.Products.ToListAsync());**

**}**

**// GET: Products/Create**

**public IActionResult Create()**

**{**

**return View();**

**}**

**// POST: Products/Create**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<IActionResult> Create([Bind("Id,Name,Price")] Product product)**

**{**

**if (ModelState.IsValid)**

**{**

**\_context.Add(product);**

**await \_context.SaveChangesAsync();**

**return RedirectToAction(nameof(Index));**

**}**

**return View(product);**

**}**

**// GET: Products/Edit/5**

**public async Task<IActionResult> Edit(int? id)**

**{**

**if (id == null)**

**{**

**return NotFound();**

**}**

**var product = await \_context.Products.FindAsync(id);**

**if (product == null)**

**{**

**return NotFound();**

**}**

**return View(product);**

**}**

**// POST: Products/Edit/5**

**[HttpPost]**

**[ValidateAntiForgeryToken]**

**public async Task<IActionResult> Edit(int id, [Bind("Id,Name,Price")] Product product)**

**{**

**if (id != product.Id)**

**{**

**return NotFound();**

**}**

**if (ModelState.IsValid)**

**{**

**try**

**{**

**\_context.Update(product);**

**await \_context.SaveChangesAsync();**

**}**

**catch (DbUpdateConcurrencyException)**

**{**

**if (!ProductExists(product.Id))**

**{**

**return NotFound();**

**}**

**else**

**{**

**throw;**

**}**

**}**

**return RedirectToAction(nameof(Index));**

**}**

**return View(product);**

**}**

**// GET: Products/Delete/5**

**public async Task<IActionResult> Delete(int? id)**

**{**

**if (id == null)**

**{**

**return NotFound();**

**}**

**var product = await \_context.Products**

**.FirstOrDefaultAsync(m => m.Id == id);**

**if (product == null)**

**{**

**return NotFound();**

**}**

**return View(product);**

**}**

**// POST: Products/Delete/5**

**[HttpPost, ActionName("Delete")]**

**[ValidateAntiForgeryToken]**

**public async Task<IActionResult> DeleteConfirmed(int id)**

**{**

**var product = await \_context.Products.FindAsync(id);**

**\_context.Products.Remove(product);**

**await \_context.SaveChangesAsync();**

**return RedirectToAction(nameof(Index));**

**}**

**private bool ProductExists(int id)**

**{**

**return \_context.Products.Any(e => e.Id == id);**

**}**

**}**

1. Lớp Product: Đây là lớp mô hình (model) đại diện cho thực thể Product với các thuộc tính Id, Name, và Price.
2. Lớp ApplicationDbContext: Đây là lớp ngữ cảnh cơ sở dữ liệu (DbContext) sử dụng Entity Framework Core để tương tác với cơ sở dữ liệu SQL Server. Nó chứa một DbSet<Product> đại diện cho bảng Products trong cơ sở dữ liệu.
3. Cấu hình kết nối cơ sở dữ liệu: Chuỗi kết nối được cấu hình trong appsettings.json và được sử dụng trong Startup.cs để thiết lập kết nối với cơ sở dữ liệu SQL Server.
4. ProductsController: Đây là bộ điều khiển (controller) chứa các hành động (actions) để thực hiện các thao tác CRUD:

* Index: Đọc và hiển thị danh sách các sản phẩm.
* Create: Tạo mới một sản phẩm.
* Edit: Chỉnh sửa thông tin sản phẩm.
* Delete: Xóa sản phẩm.
* Các hành động này tương tác với cơ sở dữ liệu thông qua ApplicationDbContext để thực hiện các thao tác CRUD.