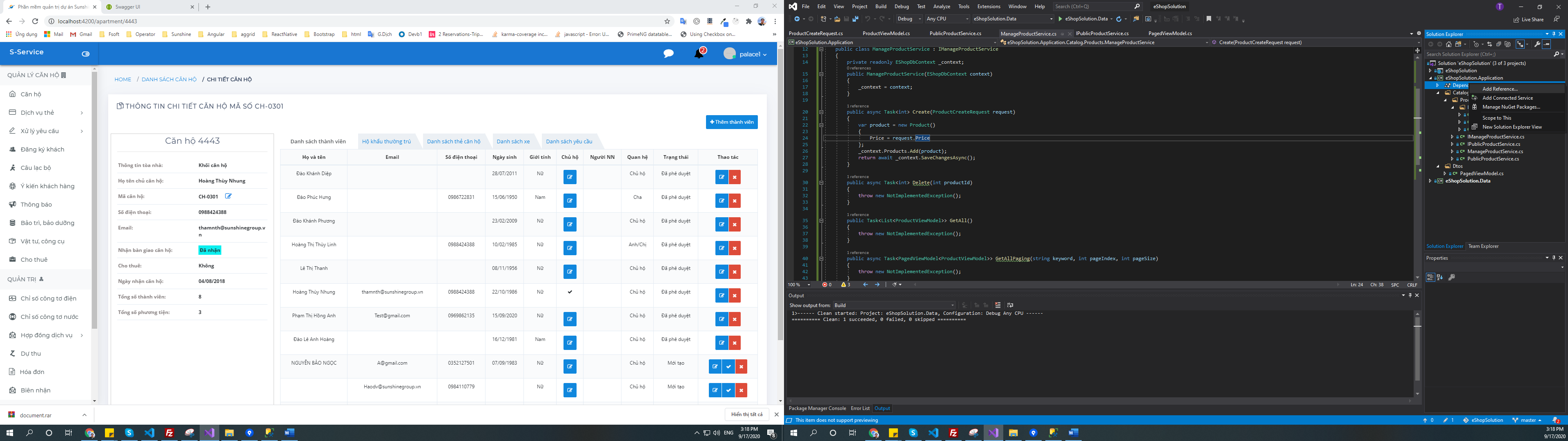
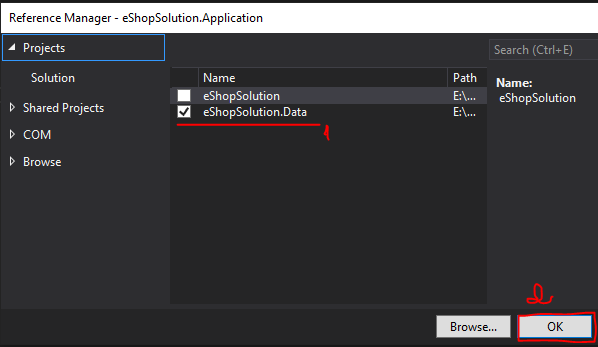
Ta chuyển sang thực hiện trên project eShopSolution.Application

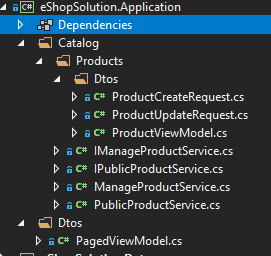
Trong project này ta sẽ gọi sang project eShopSolution.Data để lấy DbContext thao tác với cơ sở dữ liệu.

Để gọi sang được thì ta cần phải reference đến eShopSolution.Data từ eShopSolution.Application, ta thực hiện như sau:





Trong project ta thực hiện tạo ra cấu trúc thư mục như sau:



Trong đó Catelog chứa service thao tác các chức năng. Như chức năng xử lí cho product thì ta để trong thư mục Products

Thư mục Dtos ở ngoài thì chứ những file class dùng để transfer dữ liệu mapping qua lại.

Trong Dtos thì là nhưng file dùng để làm model cho Product.

Ta có 2 loại Product là PublicProduct dùng để xử lí thông tin liên quan đến Product ở trang chủ. ManageProduct dùng để xử lí các tác vụ liên quan đến Product trong trang Admin.

Trong Products thì ta tạo ra các file Interface làm quy định cho các file service thi hành. Nội dung của các file sẽ như sau:

IManageProductService.cs

using eShopSolution.Application.Catalog.Products.Dtos;

using eShopSolution.Application.Dtos;

using System.Collections.Generic;

using System.Threading.Tasks;

namespace eShopSolution.Application.Catalog.Products

{

public interface IManageProductService

{

Task<int> Create(ProductCreateRequest request);

Task<int> Update(ProductUpdateRequest request);

Task<int> Delete(int productId);

Task<List<ProductViewModel>> GetAll();

Task<PagedViewModel<ProductViewModel>> GetAllPaging(string keyword, int pageIndex, int pageSize);

}

}

Trong file này để thực hiện asynchronous cho các tác vụ ta thực hiện sử dụng Task để làm type trả lại của các function.

IPublicProductService.cs

using eShopSolution.Application.Catalog.Products.Dtos;

using eShopSolution.Application.Dtos;

namespace eShopSolution.Application.Catalog.Products

{

public interface IPublicProductService

{

public PagedViewModel<ProductViewModel> GetAllByCategoryById(int categoryId, int pageIndex, int pageSize);

}

}

Nội dung của các file Dtos sẽ như sau:

ProductCreateRequest.cs là những dữ liệu được truyền vào service dùng làm data để tạo ra product.

namespace eShopSolution.Application.Catalog.Products.Dtos

{

public class ProductCreateRequest

{

public string Name { set; get; }

public decimal Price { set; get; }

}

}

File Service sẽ có nội dung như sau:

using eShopSolution.Application.Catalog.Products.Dtos;

using eShopSolution.Application.Dtos;

using eShopSolution.Data.EF;

using eShopSolution.Data.Entities;

using System;

using System.Collections.Generic;

using System.Threading.Tasks;

namespace eShopSolution.Application.Catalog.Products

{

public class ManageProductService : IManageProductService

{

private readonly EShopDbContext \_context;

public ManageProductService(EShopDbContext context)

{

\_context = context;

}

public async Task<int> Create(ProductCreateRequest request)

{

var product = new Product()

{

Price = request.Price

};

\_context.Products.Add(product);

return await \_context.SaveChangesAsync();

}

public async Task<int> Delete(int productId)

{

throw new NotImplementedException();

}

public Task<List<ProductViewModel>> GetAll()

{

throw new NotImplementedException();

}

public async Task<PagedViewModel<ProductViewModel>> GetAllPaging(string keyword, int pageIndex, int pageSize)

{

throw new NotImplementedException();

}

public async Task<int> Update(ProductUpdateRequest request)

{

throw new NotImplementedException();

}

}

}

Ta thực hiện gọi lại context của project eShopSolution.Data ở trong constructor của class theo design pattern DI. Ta lưu lại ở một biến để thưc hiện.

Trong các hàm để thực hiện một task ta sử dụng dụng async và await

Ở đây ta thực hiện Tạo ra một Product trước.

Trong những phương thức còn lại thì ta sẽ thực hiện tương tự.

Nội dung của file DTOS ở ngoài dùng để mapping dữ liệu có nội dung như sau:

using System.Collections.Generic;

namespace eShopSolution.Application.Dtos

{

public class PagedViewModel<T>

{

public List<T> Items { get; set; }

public int TotalRecord { set; get; }

}

}