Đầu tiên ta thực hiện sửa lại file EShopDbContext.cs kế thừa lại từ IdentityDbContext (Ta cài đặt từ nuget trước Microsoft.AspNetCore.Identity)

Ta tạo các file với nội dung như sau (Những file này nếu muốn thêm thông tin thì tạo, còn không sử dụng lại của Identity)

Trong thư mục eShopSolution.Data > Entities:

AppRole.cs

using Microsoft.AspNetCore.Identity;

using System;

namespace eShopSolution.Data.Entities

{

public class AppRole: IdentityRole<Guid>

{

public string Description { get; set; }

}

}

AppUser.cs

using Microsoft.AspNetCore.Identity;

using System;

namespace eShopSolution.Data.Entities

{

public class AppUser: IdentityUser<Guid>

{

public string FirstName { set; get; }

public string LastName { get; set; }

public DateTime Dob { get; set; }

}

}

Trong eShopSolution.Data > Configurations:

AppRoleConfiguration.cs

using eShopSolution.Data.Entities;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata.Builders;

namespace eShopSolution.Data.Configurations

{

public class AppRoleConfiguration : IEntityTypeConfiguration<AppRole>

{

public void Configure(EntityTypeBuilder<AppRole> builder)

{

builder.ToTable("AppRoles");

builder.Property(x => x.Description).HasMaxLength(250).IsRequired();

}

}

}

AppUserConfiguration.cs

using eShopSolution.Data.Entities;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata.Builders;

namespace eShopSolution.Data.Configurations

{

public class AppUserConfiguration : IEntityTypeConfiguration<AppUser>

{

public void Configure(EntityTypeBuilder<AppUser> builder)

{

builder.ToTable("AppUsers");

builder.Property(x => x.FirstName).HasMaxLength(100);

builder.Property(x => x.LastName).HasMaxLength(100);

builder.Property(x => x.Dob).IsRequired();

}

}

}

Tiếp theo ta thực hiện thêm Relationship cho bảng AppUser như sau:

1 User có nhiều Transaction

1 User có nhiều Order

1 User có nhiều Cart

Trong class AppUser.cs ta thêm như sau:

public class AppUser: IdentityUser<Guid>

{

public List<Cart> Carts { get; set; }

public List<Order> Orders { get; set; }

public List<Transaction> Transactions { get; set; }

}

Trong Cart.cs ta thêm quan hệ một tới User:

public class Cart

{

public AppUser AppUser { get; set; }

}

Trong Transaction.cs ta thêm quan hệ tới bảng AppUser:

public class Transaction

{

public AppUser AppUser { get; set; }

}

Trong CartConfiguration.cs ta thêm cấu hình quan hệ:

public void Configure(EntityTypeBuilder<Cart> builder)

{

builder.HasOne(x => x.AppUser).WithMany(x => x.Carts).HasForeignKey(x => x.UserId);

}

Trong OrderConfiguration.cs ta thêm cấu hình quan hệ:

public void Configure(EntityTypeBuilder<Order> builder)

{

builder.HasOne(x => x.AppUser).WithMany(x => x.Orders).HasForeignKey(x => x.UserId);

}

Trong TransactionConfiguration.cs ta thêm cấu hình quan hệ:

public void Configure(EntityTypeBuilder<Transaction> builder)

{

builder.HasOne(x => x.AppUser).WithMany(x => x.Transactions).HasForeignKey(x => x.UserId);

}

}

Trong file EShopDbContext.cs thêm đoạn config của 2 bảng vừa thêm và các bảng còn lại ta được:

Trong method OnModelCreating:

modelBuilder.ApplyConfiguration(new AppUserConfiguration());

modelBuilder.ApplyConfiguration(new AppRoleConfiguration());

modelBuilder.Entity<IdentityUserClaim<Guid>>().ToTable("AppUserClaims");

modelBuilder.Entity<IdentityUserRole<Guid>>().ToTable("AppUserRoles").HasKey(x => new { x.UserId, x.RoleId});

modelBuilder.Entity<IdentityUserLogin<Guid>>().ToTable("AppUserLogins").HasKey(x => x.UserId);

modelBuilder.Entity<IdentityRoleClaim<Guid>>().ToTable("AppUserRoleClaims");

modelBuilder.Entity<IdentityUserToken<Guid>>().ToTable("AppUserTokens").HasKey(x => x.UserId);

Những bảng IdentityUserClain,IdentityUserRole,… ta sử dụng lại nguyên thuộc tính đã có sẵn nên ta không cần phải tạo ra những class mới và configuration như AppUser và AppRole

Ta thực hiện thêm Seeding cho identity như sau, trong file Extensions > ModelBuilderExtensions.cs

Ta thêm như sau:

public static void Seed(this ModelBuilder modelBuilder)

{

// Seeding identity data

var roleId = new Guid("AAAABA55-808E-479F-BE8B-72F69913442F");

var adminId = new Guid("DDD4BA55-808E-479F-BE8B-72F69913442F"); modelBuilder.Entity<AppRole>().HasData(

new AppRole { Id = roleId, Name = "admin", NormalizedName = "admin", Description = "Description Role"});

var hasher = new PasswordHasher<AppUser>();

modelBuilder.Entity<AppUser>().HasData(

new AppUser

{

Id = adminId,

UserName = "admin",

NormalizedUserName = "admin",

Email = "tano@gmail.com",

NormalizedEmail = "tano@gmail.com",

EmailConfirmed = true,

PasswordHash = hasher.HashPassword(null, "123456a@"),

SecurityStamp = string.Empty,

FirstName = "Tano",

LastName = "Tano",

Dob = new DateTime(1996, 07, 14)

});

modelBuilder.Entity<IdentityUserRole<Guid>>().HasData(new IdentityUserRole<Guid>

{

RoleId = roleId,

UserId = adminId

});

}

Tiếp theo trong Package Manager Console ta thêm migration cho những cấu hình mà ta vừa thêm để cập nhật lại cơ sở dữ liệu:

Add-migration

Update-database