**1.创建模型类**

* 打开.Core项目,新建新建一个项目文件夹(Clothes);  
  为了演示表关联及外键的使用，创建两个类：
* 创建类ClothesCategoty.cs

using Abp.Domain.Entities;

namespace myTest.Clothes

{

public class ClothesCategory:Entity

{

public virtual string Name { get; set; }

}

}

using Abp.Domain.Entities引用Abp.Domain.Entities，abp中所有的类都继承自Abp.Domain.Entities.Entity,集成后会自动创建表主键字段Id。

* 创建类Clothes.cs

using System;

using System.ComponentModel.DataAnnotations.Schema;

using Abp.Domain.Entities;

namespace myTest.Clothes

{

[Table("Clothes")]

public class Clothes:Entity

{

public virtual DateTime CreationTime { get; set; }

public virtual string PictureUrl { get; set; }

[ForeignKey("ClothesCategoryId")]

public virtual ClothesCategory ClothesCategory { get; set; }

}

}

[Table("xxx")]指定表名，不指定默认使用类名；[ForeignKey("xxx")]指定关联表外键的名称。

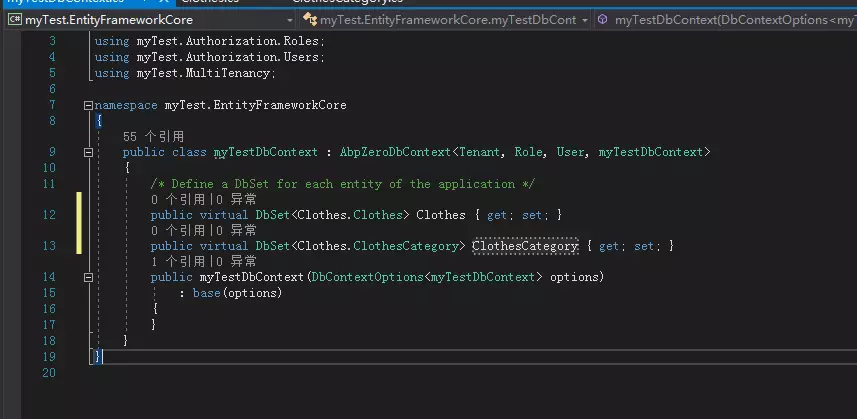
**2.添加类到DbContext**

打开.EntityFrameworkCore项目，找到xxxDbContext类，在类中加入新模型的相关代码：

public virtual DbSet<Clothes.Clothes> Clothes { get; set; }

public virtual DbSet<Clothes.ClothesCategory> ClothesCategory { get; set; }

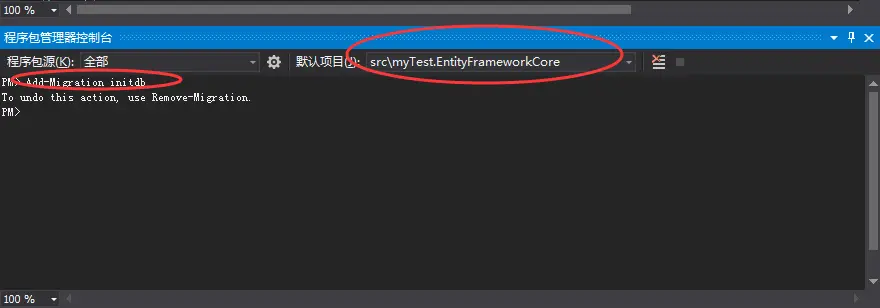
如下图：



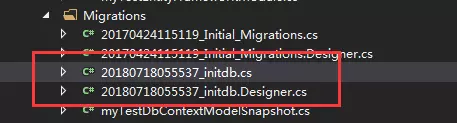
**3.执行命令**

打开NuGet程序包管理控制台，选择默认项目为.EntityFrameworkCore项目。

* 执行Add-Migration xxxx,其中xxxx可自主命名；



执行完成后会在项目的Migrations文件夹下自动生成两个文件，如下图：



* 执行“Update-Database”

**至此，数据库表创建完成。**

**4.数据库的更新**

如果出现了字段的新增，则在core中的类中新增对应的字段，再按照上面第三点执行相关命令映射到数据库中。

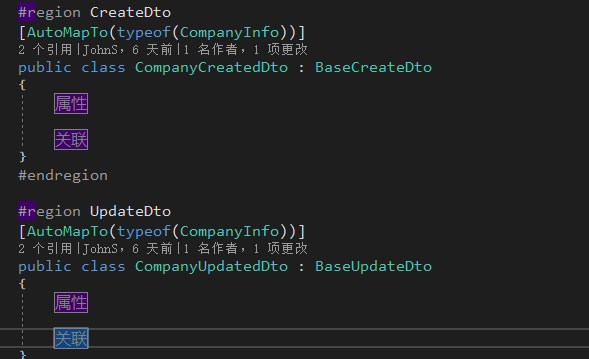
**======至此，实体类MODEL的字段就映射到数据库中了===========**

**5.Application层创建增删改查**

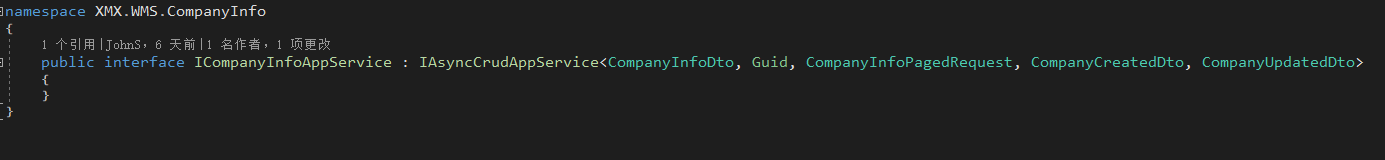
a.在application层创建对应的文件夹



b.加入查询、新增、更新的DTO

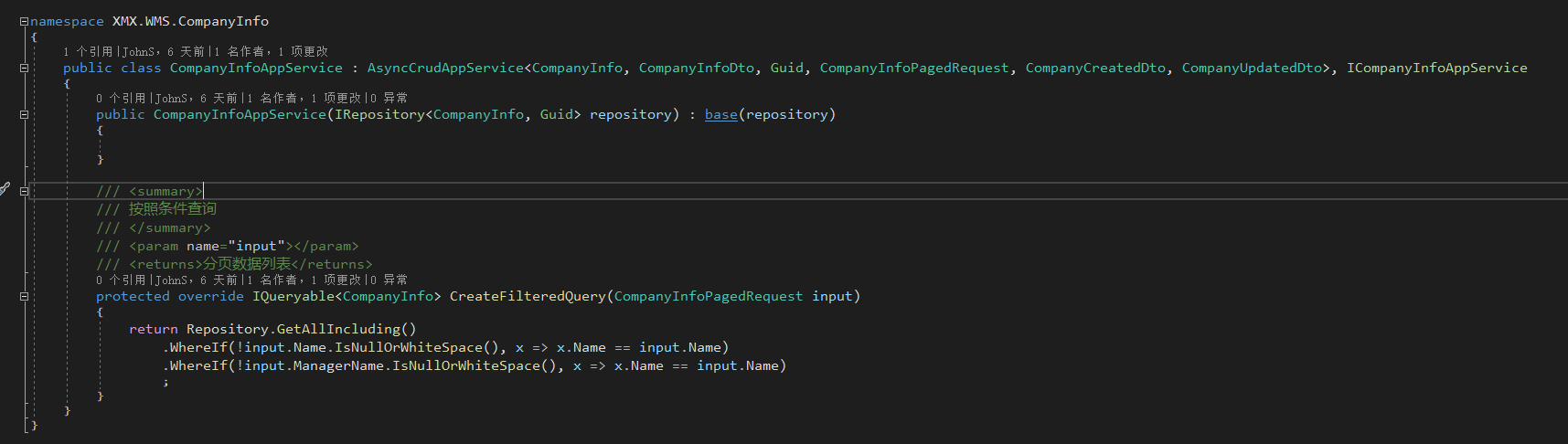


c.增加Service里面的接口的代码



其中，IAsyncCrudAppService用于生成增删改查的接口

d.Service里面的代码



Service里面继承自上面的接口，其中CreateFilteredQuery用于查询的过滤

按照这个就可以生成对应的基础的增删改查。

**6.权限**

1、.Core -> Authorization -> PermissionNames.cs 这里新增权限项，可以具体到按钮（即方法名）

public const string Pages\_Users\_ChangeOnline = “Pages.Users.ChangeOnline”;

2、.Core》Authorization》FrameworkAuthorizationProvider.cs 这里创建权限，这样在页面中添加角色的时候才能显示出来，项目运行之后，程序会自动给管理员的1.2角色创建对应的菜单

context.CreatePermission(PermissionNames.Pages\_Users\_ChangeOnline, L("ChangeOnline"));

3、具体方法的权限控制

3.1、方法上，加上[AbpAuthorize(PermissionNames.Pages\_Users\_ChangeOnline)]

3.2、类上加， [AbpAuthorize(PermissionNames.Pages\_Users)]

4、 创建带有该按钮功能的角色，或者修改角色，使角色带有该菜单的功能

看Role/Create或者 Role/Update 中的grantedPermissions中填上要赋予的权限菜单就可以