## **一、领域服务的引入**

有些重要的领域操作，不适合归到实体(Entity)和值对象(Value Object)的类别中，这些操作从本质上讲是有些活动或动作，而不是事物，**当领域中的某个重要过程或转换操作不属于实体或值对象的自然职责时，应该在模型中添加一个作为独立接口的操作，并将其申明为Service**。

## **二、领域服务的使用**

还记得我们介绍仓储的时候我们领域层里面的Services文件夹么？下面，我们就根据给指定的用户赋予制定的权限的功能一步一步写一个领域服务功能。

### **1、聚合划分的修改**

这里需要提出一个问题，对于博主在[C#进阶系列——DDD领域驱动设计初探（一）：聚合](http://www.cnblogs.com/landeanfen/p/4816706.html)这篇里面聚合的划分，现在想来存在不合理的地方，上次说了应该划分为4个聚合 ，可是没有考虑到用户和角色是多对多的关系，所以需要将用户角色表TB\_USERROLE单独划分为一个聚合，所以总共是5个聚合，这也就是前面说的对于DDD里面聚合的划分是比较考量程序员经验的，对于聚合划分有误造成的误解表示抱歉。我需要在相应的地方做下修改。

领域实体要继承聚合根基类：

public partial class TB\_USERROLE:AggregateRoot

{

}

新建仓储接口和仓储实现

public interface IUserRoleRepository : IRepository<TB\_USERROLE>

{

}

[Export(typeof(IUserRoleRepository))]

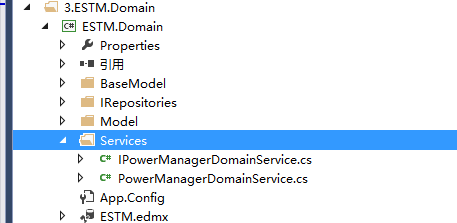
public class UserRoleRepository : EFBaseRepository<TB\_USERROLE>, IUserRoleRepository

{

}

### **2、领域服务代码的搭建**

在领域层的Services文件夹建立服务的接口和实现



public interface IPowerManagerDomainService

{

void AssignPower(TB\_USERS oUser, TB\_ROLE oRole);

}

　　[Export(typeof(IPowerManagerDomainService))]

public class PowerManagerDomainService:IPowerManagerDomainService

{

private IUserRepository \_userRepository = null;

private IRoleRepository \_roleRepository = null;

private IUserRoleRepository \_userroleRepository = null;

[ImportingConstructor]

public PowerManagerDomainService(IUserRoleRepository oUserRoleRepository)

{

\_userroleRepository = oUserRoleRepository;

}

public void AssignPower(TB\_USERS oUser, TB\_ROLE oRole)

{

if (oUser == null || oRole == null)

{

return;

}

var oUserRole = \_userroleRepository.Find(x => x.USER\_ID == oUser.USER\_ID && x.ROLE\_ID == oRole.ROLE\_ID).FirstOrDefault();

if (oUserRole == null)

{

oUserRole = new TB\_USERROLE();

oUserRole.ROLE\_ID = oRole.ROLE\_ID;

oUserRole.USER\_ID = oUser.USER\_ID;

oUserRole.ID = Guid.NewGuid().ToString();

\_userroleRepository.Insert(oUserRole);

}

}

}

### **3、调用测试**

在应用层里面我们来写测试代码：

　　 class Program

{

[Import(AllowDefault=false,AllowRecomposition=true,RequiredCreationPolicy=CreationPolicy.Any,Source=ImportSource.Any)]

public IPowerManagerDomainService powerDomainService { get; set; }

static void Main(string[] args)

{

var oProgram = new Program();

Regisgter.regisgter().ComposeParts(oProgram);

var oUser = new TB\_USERS() { USER\_ID = "04acd48a819447d388b20dffb15f672e" };

var oRole = new TB\_ROLE() { ROLE\_ID = "cccc" };

oProgram.powerDomainService.AssignPower(oUser, oRole);

Console.ReadKey();

}

}