**IObjectMapper 接口**

IObjectMapper是一个简单的包含把一个对象映射到另一个对象的方法的抽象，我们可以用如下代码书写上例：

public class UserAppService : ApplicationService

{

**private readonly IObjectMapper \_objectMapper;**

public UserAppService(**IObjectMapper objectMapper**)

{

\_objectMapper = objectMapper;

}

public void CreateUser(CreateUserInput input)

{

**var user = \_objectMapper.Map<User>(input);**

}

}

**集成 AutoMapper**

**安装 ( 启动模板下载的项目已安装 )**

首先,安装[Abp.AutoMapper](https://www.nuget.org/packages/Abp.AutoMapper" \t "http://www.cnblogs.com/kid1412/p/_blank)nuget包到你的项目里：

Install-Package Abp.AutoMapper

然后，在你的模块上方添加对**AbpAutoMapperModule** 的依赖：

**[DependsOn(typeof(AbpAutoMapperModule))]**

public class MyModule : AbpModule

{

}

**直接映射**

memberCouponDtos = AutoMapper.Mapper.Map<List<MemberCouponDto>>(memberCouponEntityModels)

**自动映射的特性**

**[AutoMapTo(typeof(User))]**

public class CreateUserInput

{

public string Name { get; set; }

public string Surname { get; set; }

public string EmailAddress { get; set; }

public string Password { get; set; }

}

AutoMap特性在两个类之间双向映射，但在这个示例里，我们只需要从CreateUserInput映射到User,所以我们只需要用AutoMapTo

**自定义映射**

[DependsOn(typeof(AbpAutoMapperModule))]

public class MyModule : AbpModule

{

public override void PreInitialize()

{

Configuration.Modules.AbpAutoMapper().Configurators.Add(config =>

{

**config.CreateMap<CreateUserInput, User>()**

**.ForMember(u => u.Password, options => options.Ignore())**

**.ForMember(u => u.Email, options => options.MapFrom(input => input.EmailAddress));**

});

}

}