

در پست قبلی مقدمه ای داشتیم بر AutoMapper؛ مثالی که در اون پست عنوان شد ساده ترین و پرکاربردترین روش استفاده از AutoMapper هست بنام Flattening که در واقع از یک شیء کل به یک شیء کوچکتر می رسیم. همانطور که در قسمت اول گفتیم AutoMapper کارش رو بر اساس قراردادهای انجام میدی یا همون Convention Base. یکی از قراردادهای AutoMapper، نگاشت براساس نام اعضای اون شیء هست؛ مثلاً در مثال قبلی FirstName در مبداء، به خاصیتی با همین نام نگاشت شد و ...

Projection

روش استفاده بعدی که به اون Projection (معنی فارسی خوب برا Projection چیه ؟) میگن برای مواقعی هست که اعضای یک شیء در مبداء، همتایی در مقصد ندارد (از نظر نام) و در واقع می خواهیم یک شیء رو به یک شیء دیگه تغییر شکل بدیم. مدل زیر رو در نظر بگیرید:

```
public class Person
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}
```

که قرار است به مدل PersonDTO نگاشت بشه.

```
public class PersonDTO
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Family { get; set; }
    public string FullName { get; set; }
    public string Email { get; set; }
}
```

همانطور که می بینید در مقصد خاصیت هایی داریم که در مبداء همتایی ندارند. برای نگاشت چنین اشیایی، از متد ForMember استفاده و نگاشت های شیء های مورد نظر رو Custom می کنیم.

```
Mapper.CreateMap<Person, PersonDTO>().ForMember(des => des.Name, op => op.MapFrom(src =>
src.FirstName)).
    ForMember(des => des.FullName, op => op.MapFrom(src => src.FirstName + " " +
src.LastName)).ForMember(
    des => des.Email, op => op.Ignore());
```

متد ForMember از یک Action Delegate برای کانفیگ کردن هر عضو استفاده میکنه. در مثال بالا ما از MapFrom برای اعمال نگاشت Custom استفاده کردیم.

```
AutoMapper.IMappingExpression<TSource,TDestination>.ForMember(System.Linq.Expressions.Expression<System.Func<TDestination,object>>, System.Action<AutoMapper.IMemberConfigurationExpression<TSource>>)
```

نکته: برای تست کردن اینکه آیا نگاشت ما Exception داره یا نه میتوان از متد زیر استفاده کرد.

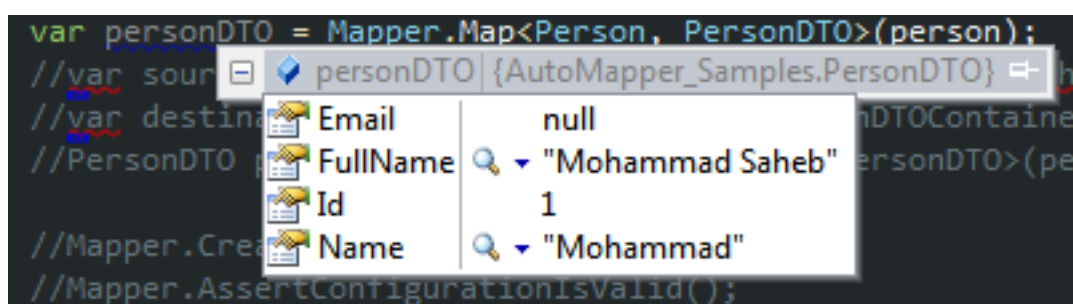
```
Mapper.AssertConfigurationIsValid();
```

نکته : همیشه موقع نگاشت، اعضای شیء مقصد برای AutoMapper مهمن؛ مثلاً در مثال بالا خاصیت LastName در مبداء به هیچ عضوی در مقصد نگاشت نشده (به صورت مستقیم) و این تولید Exception نمیکند ولی برعکس اون باعث تولید Exception میشه؛ مثلاً اگر خاصیت Email رو که در مبداء همتایی نداره رو کانفیگ نکنیم، باعث تولید Exception میشه.

نحوه استفاده

```
var person = new Person
{
    Id = 1,
    FirstName = "Mohammad",
    LastName = "Saheb",
};
var personDTO = Mapper.Map<Person, PersonDTO>(person);
```

خروجی به شکل زیر میشه



Collection

بعد از نوشتن کانفیگ نگاشت‌ها، بدون نیاز به تنظیمات خاصی میتونیم مجموعه ای از شی‌های مبداء رو به مقصد نگاشت کنیم. مجموعه‌های پشتیبانی شده به شرح زیرن.

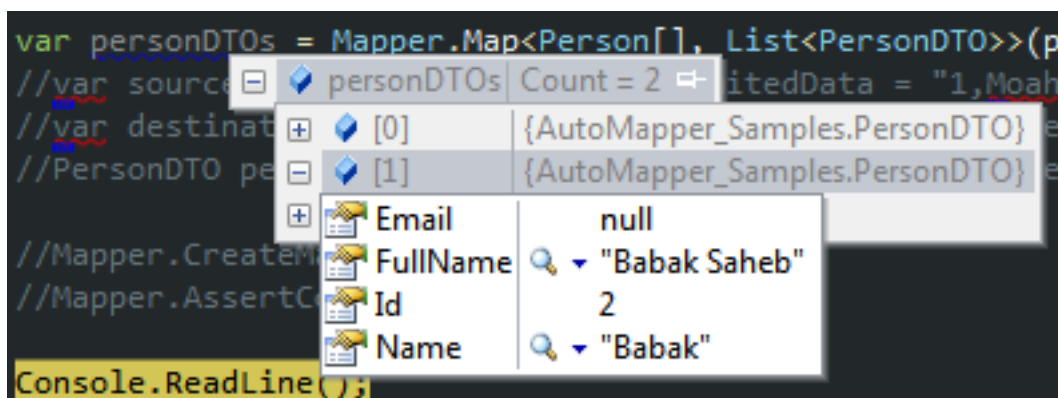
```
IEnumerable, IEnumerable<T>, ICollection, ICollection<T>, IList, IList<T>, List<T>.
```

و البته آرایه ها.

مثال

```
var persons = new[]
{
    new Person { Id = 1, FirstName = "Mohammad", LastName = "Saheb" },
    new Person { Id = 2, FirstName = "Babak", LastName = "Saheb" }
};
var personDTOs = Mapper.Map<Person[], List<PersonDTO>>(persons);
```

خروجی به شکل زیر میشه



Nested Mappings

برای نگاشت کلاس های تو در تو از این روش استفاده می کنیم و ...

فرض کنید در کلاس Person خاصیتی از نوع Address داریم و در کلاس PersonDTO خاصیتی از نوع AddressDTO.

```

public class Address
{
    public string Ad1 { get; set; }
    public string Ad2 { get; set; }
}

public class AddressDTO
{
    public string Ad1 { get; set; }
    public string Ad2 { get; set; }
}

```

برای نگاشت هایی از این دست باید تنظیمات نگاشت مربوط به نوع های تودرتو را صریحا معین کنیم.

```
Mapper.CreateMap<Address, AddressDTO>();
```

نکته: هرچند منطقی تر بنظر میرسد که تعریف نگاشت های داخلی ابتدا بیاد، ولی فرقی نمیکند که تعریف این نگاشت قبل یا بعد از نگاشت اصلی باشد.
ادامه دارد...

نظرات خوانندگان

نویسنده: ابراهیم
تاریخ: ۲۱:۴۱ ۱۳۹۱/۰۴/۲۲

ممنون از مطالب خوبتان. Projection را به صورت پرتو در درس پایگاه داده استاد حق جو شنیده ام و به نظرم ترجمه خوبی است.

نویسنده: فرهاد یزدان پناه
تاریخ: ۱:۳ ۱۳۹۲/۰۳/۱۰

وقت بخیر جناب صاحب
در حالاتی که از AutoMapper در یک عبارت LINQ استفاده بشه در صورت وجود مواردی همچون FullName در مقصد خطای "Object reference not set to an instance of an object" رو throw می کنه.
آیا شما هم با چنین مشکلی برخورد کردید؟

نویسنده: محمد صاحب
تاریخ: ۱۰:۴۶ ۱۳۹۲/۰۳/۱۱

نمیدونم شاید. برای بررسی بیشتر کدهاش رو بذار

نویسنده: لیلا
تاریخ: ۲۳:۱۶ ۱۳۹۲/۰۹/۰۴

با تشکر
به چه روشی می توان یک vieamodel که ترکیبی از 3 model هست به مدل های مربوطه به وسیله AutoMapper ، نگاشت کرد

نویسنده: امیر هاشم زاده
تاریخ: ۱:۴۳ ۱۳۹۴/۰۳/۱۱

از روش بیان شده در [این لینک](#) استفاده کنید