

در ادامه قسمت [قبلی](#) به بررسی ویژگی های پیشرفته ی AutoMapper می پردازیم...

Custom type converters

همانطور که از اسمش مشخصه، زمانی کاربرد داره که نوع عضو یا اعضای یک شی در مبداء، با معادلشون در مقصد یکی نیستند. مثلا فرض کنید نوع Bool در مبداء رو می خواهیم به نوع String در مقصد نگاشت کنیم؛ همون Yes و No معروف بجای True یا False. کلاس های زیر رو در نظر بگیرید:

```
public class Source
{
    public string Value1 { get; set; }
    public string Value2 { get; set; }
    public string Value3 { get; set; }
}

public class Destination
{
    public int Value1 { get; set; }
    public DateTime Value2 { get; set; }
    public Type Value3 { get; set; }
}
```

طبق [مستندات](#) AutoMapper اگه بخواهیم این دو رو نگاشت کنیم Exception میده چون AutoMapper نمیدونه چطوری باید مثلا Int رو به String تبدیل کنه؛ برای همین ما باید به AutoMapper بگیم چطور این تبدیل نوع رو انجام بده.

نکته: در تستی که من انجام دادم، AutoMapper تبدیل نوع های ابتدایی رو خودش انجام میده؛ مثلا همین تبدیل Int به String رو!

یکی از روش های مهیا کردن تبدیل کننده ی نوع، پیاده سازی اینترفیس `ITypeConverter<TSource, TDestination>` هست. تقریبا مثل کاری که در WPF و SL با پیاده سازی اینترفیس `IValueConverter` انجام می دادیم. من برای تست از همون تبدیل نوع Bool به String استفاده میکنم و البته بخاطر ساده بودن دیگه Model ها رو نمی نویسم. ابتدا تعریف کلاس تبدیل کننده ی نوع:

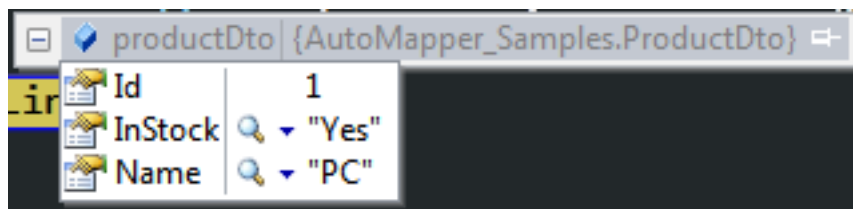
```
public class BoolToStringTypeConvertor : ITypeConverter<bool, string>
{
    public string Convert(ResolutionContext context)
    {
        return (bool)context.SourceValue ? "Yes" : "No";
    }
}
```

و نحوه استفاده:

```
Mapper.CreateMap<bool, string>().ConvertUsing<BoolToStringTypeConvertor>();
Mapper.CreateMap<Product, ProductDto>();
Mapper.AssertConfigurationIsValid();

var product = new Product { Id = 1, Name = "PC", InStock = true };
var productDto = Mapper.Map<Product, ProductDto>(product);
```

خروجی به شکل زیر میشه.



نکته: TypeConverter ها میدان دیدشون سراسریه و نیازی نیست به ازای هر نگاشتی اونو به AutoMapper معرفی کنیم Global Scope.

Custom value resolvers

کلاس های زیر رو در نظر بگیرید

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

public class PersonDTO
{
    public int Id { get; set; }
    public string RawData { get; set; }
}
```

فرض کنید داخل RawData تمامی اعضای شی مبداء رو به صورت Comma Delimited ذخیره کنیم. برای این کار می تونیم از Value Resolver استفاده کنیم.

یک روش برای این کار ارث بری از کلاس Abstract ی بنام `ValueResolver<TSource, TDestination>` هست.

```
public class CommaDelimitedResolver:ValueResolver<Person,string>
{
    protected override string ResolveCore(Person source)
    {
        return string.Join(",", source.Id, source.FirstName, source.LastName);
    }
}
```

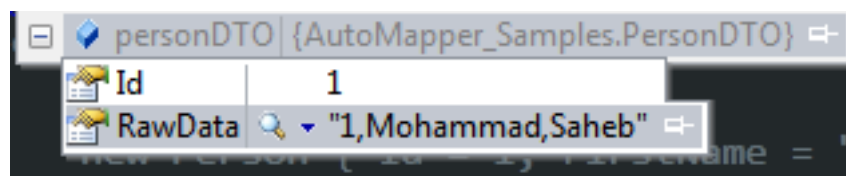
و نحوه استفاده

```
Mapper.CreateMap<Person, PersonDTO>().ForMember(
    des => des.RawData, op => op.ResolveUsing<CommaDelimitedResolver>());

var person = new Person
{
    Id = 1,
    FirstName = "Mohammad",
    LastName = "Saheb",
};

var personDTO = Mapper.Map<Person, PersonDTO>(person);
```

و خروجی به شکل زیر میشه



نکته: توجه کنید این فقط یک مثال بود و این کار رو با روش های دیگه هم میشه انجام داد مثلا MapFrom و...
نکته: میدان دید Value Resolver ها سراسری نیست و باید به ازای هر نگاشتی اونو معرفی کنیم.

Custom Value Formatters

فرض کنید تاریخ رو در بانک، به صورت میلادی ذخیره کرده اید و می خواهید سمت View به صورت شمسی نمایش بدید. بنابراین در مبدا ویژگی بنام MiladiDate از نوع DateTime دارید و در مقصد ویژگی بنام ShamsiDate از نوع String. هنگام نگاشت، AutoMapper به صورت پیش فرض ToString رو فراخونی میکنه که بدرد ما نمیخوره و...
 برای این کار میشه از Value Formatter استفاده کرد با پیاده سازی اینترفیس IValueFormatter.

```
public class ShamsiFormatter:IValueFormatter
{
    public string FormatValue(ResolutionContext context)
    {
        return ToShamsi(context.SourceValue.ToString());
    }
}
```

نحوه استفاده

```
Mapper.CreateMap<Person, PersonDTO>().ForMember(
    des => des.ShamsiDate, op => op.AddFormatter<ShamsiFormatter>());
```

نظرات خوانندگان

نویسنده: torisoft
تاریخ: ۲:۱۴ ۱۳۹۱/۰۵/۰۹

سلام جناب صاحب
بیخشید من یه کاربر مبتدیم سوالم رو خیلی ساده میگم.
من یه پروژه سیلور mvvm دارم با سرویس web api
اول اینکه Automapper تو سیلور به چه صورت عمل میکنه ؟
دوم اینکه مدل های اصلی سمت سرور نوشته میشن. حالا این مدل ها باید سمت سیلور نوشته بشن ؟
سوم اینکه امکان نگاشت دو یا چند مدل به صورت همزمان به یک DTO وجود دارد ؟
چهارم اینکه تعریف نگاشت باید در سمت سرور باشد یا سیلور ؟ به عبارت دیگه در کنترلر سمت سرور یا در viewmodel سمت سیلور یا هیچکدام ؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۸:۴۵ ۱۳۹۱/۰۵/۰۹

در سمت کلاینت سیلور لایت به صورت رسمی [پشتیبانی نمی شود](#) . سمت سرور آن هم همین مواردی است که تا الان گفته شده و زمانیکه دات نت فول در اختیار شما باشد مباحث آن یکی است و تفاوتی نمی کند.

نویسنده: محمد صاحب
تاریخ: ۱۰:۳۸ ۱۳۹۱/۰۵/۰۹

با تشکر از آقای نصیری...
3-بله میشه که بهش [Flattening](#) میگن.

نویسنده: مسعود
تاریخ: ۲:۳۲ ۱۳۹۱/۰۵/۱۰

جناب نصیری و صاحب سلام
به چه صورت اطلاعات رو از context گرفته و map کنیم ؟
منظورم اینکه کد زیر بدون استفاده از automapper درسته

```
private TollContext db = new TollContext();  
  
private IQueryable<CarDTO> MapCars()  
{  
    return from p in db.Cars select new CarDTO() { Id = p.Id, Name = p.Name, Price = p.Price };  
}  
  
public IEnumerable<CarDTO> GetCars()  
{  
    return MapCars().AsEnumerable();  
}
```

ولی با استفاده از automapper (البته نمی دونم به چه صورته) به شکل زیر نوشتم که غلطه

```
private TollContext db = new TollContext();  
  
private IQueryable<CarDto> MapCars()  
{  
    return Mapper.Map<Car, CarDto>(db.Cars);  
}  
  
public IEnumerable<CarDto> GetCars()
```

```
{  
    return MapCars().AsEnumerable();  
}
```

لطفا در صورت امکان راهنمایی بفرمائید.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۰ ۸:۲۹

[این پیشنهاد](#) رو اول مطالعه کنید.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۶/۰۲ ۱۹:۴۲

با سلام. لطفا اگر امکانش هست در مورد مپ کردن چند نوع داده ای مبدا به یک نوع داده ای مقصد هم توضیح دهید. ممنونم.
مثال:

```
public class SourceType1 //poco class  
{  
    public string Name  
    public string JobTitle  
    public string PetsName  
}  
  
public class SourceType2  
{  
    public string BankName  
    public decimal CreditCardBalance  
    public DateTime SignupDate  
}  
  
public class Destination  
{  
    public string Job  
    public string Balance //thousand separator  
}
```

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۶/۰۳ ۱۴:۱۴

ممنون...

این مورد برا من پیش نیومده بود.
معمولا به این صورته که ما یک کلاس مرکب داریم و از اون میرسیم به کلاس سبک تر یا همون [Flattening](#) و البته یک روش برای حل مشکل شما هم همینه اون 2 تا کلاس مبدا رو تو یه کلاس مرکب داشته باشید و...
یا اینکه از این کلاس [کمکی](#) که اینجا معرفی شده استفاده کنید.
نحوه فراخونی به این شکل میشه

```
var Destination = EntityMapper.Map<Destination>(SourceType1, SourceType2);
```

نویسنده: میثم
تاریخ: ۱۳۹۱/۱۰/۱۵ ۲۰:۲۱

سلام ، مشکلی که من با Automapper دارم اینه
کلاسی به شکل زیر دارم

```
public class ProductType: BaseEntity
```

```

{
    #region Field
    private string persianTitle;
    private string englishTitle;
    private IList<Product> products;
    #endregion
    #region Memebr
    public string PersianTitle
    {
        get
        {
            return (persianTitle);
        }
        set
        {
            persianTitle = Microsoft.Security.Application.Encoder.HtmlEncode(value);
        }
    }
    public string EnglishTitle
    {
        get
        {
            return (englishTitle);
        }
        set
        {
            englishTitle = Microsoft.Security.Application.Encoder.HtmlEncode(value);
        }
    }
    public virtual IList<Product> Products
    {
        get
        {
            return (products);
        }
        set
        {
            products = value;
        }
    }
    #endregion
}

```

و خوب کلاسی با عنوان Product هم موجوده با کدی به این شکل سعی در آپدیت کردن این کلاس داریم

```

[HttpPost]
public ActionResult Update(ProductTypeViewModel productTypeViewModel)
{
    if (productTypeViewModel.ExaminId())
    {
        ProductType productType;
        productType = _productType.Find(x => x.Id == productTypeViewModel.Id);
        AutoMapper.Mapper.Map(productTypeViewModel, productType);
        _uow.SaveChanges();
    }
    return RedirectToAction("Index");
}

```

متأسفانه خطا رخ میدهد ، خطایی مربوط به Context ظاهراً اینم پیام خطا

issing type map configuration or unsupported mapping.

Mapping types:

ProductTypeViewModel -> ProductType_5334DF7BAFBE780DF5328E2D6DF2A0DC3350F23340BE2EE2FC506AE9EDEC38DA
 MvcUserInterface.Areas.Management.Models.ProductTypeViewModel ->
 System.Data.Entity.DynamicProxies.ProductType_5334DF7BAFBE780DF5328E2D6DF2A0DC3350F23340BE2EE2FC506AE9EDEC38DA

Destination path:

ProductType_5334DF7BAFBE780DF5328E2D6DF2A0DC3350F23340BE2EE2FC506AE9EDEC38DA

Source value:

MvcUserInterface.Areas.Management.Models.ProductTypeViewModel

ممنون میشم اگه کسی تجربه ای داره کمکم کنه من برای پیاده سازی لایه های سرویس ، دامین و دیتا از روشی که آقای نصیری گفته استفاده کردم . ویرایش کلاس هایی که دارای عضوی به صورت لیست از کلاس دیگر هستند چگونه باید باشه ؟ برای map کردن این کلاس ها کار خاصی باید انجام بشه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۵ ۲۱:۳

این نام کلاس های طولانی رو که مشاهده می کنید در حقیقت پشت صحنه EF است و کلاس های پروکسی نام دارند. بنابراین نیاز به کمی تنظیم بیشتر هست. [ادامه در اینجا](#)

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۰/۲۸ ۰:۵۲

آیا امکان تعریف AutoMapper در لایه های دیگر هم وجود دارد ؟ در صورت مثبت بودن پاسخ، چگونه اون رو توی لایه UI صدا بزنیم ؟

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۲۸ ۸:۵۰

کار automapper نگاشت خواص لایه ui به domain برنامه است و برعکس. اینکار در mvc مثلا در کنترلرها انجام میشه. کنترلر هم نتیجه کار رو با return View به لایه نمایشی برای استفاده ارسال می کنه.

نویسنده: ali
تاریخ: ۱۳۹۱/۱۲/۰۱ ۱۱:۲۷

attribute های مدل مانند Display را چرا وقتی Map میکنیم نمیاره؟

به طور مثال در صورتی میاره که به شکل زیر باشه

```
public class Customer
{
    public Customer()
    {
        Orders = new List<Order>();
    }
    [StringLength(10)]
    public string Title { get; set; }

    [Display(Name = "نام")]
    public string FirstName { get; set; }

    [Display(Name = "نام خانوادگی")]
    public string LastName { get; set; }
    public ICollection<Order> Orders { get; set; }
}
```

```
public class CustomerViewModel
{
    public Customer Customer { get; set; }
}
```

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۱۲/۰۱ ۱۴:۳۴

گویا این [امکان](#) موجود نیست من که خودم Data Annotation ها رو تو ViewModel تکرار میکنم. برای بررسی بیشتر لطفا کد قسمت کانفیگ Mapping و همچنین کلاسها تون رو بنویسید.

نویسنده: ali
تاریخ: ۱۳۹۱/۱۲/۰۷ ۱۳:۰۹

map کرد میشه دو طرفه باشه.

به طور مثال :

```
Mapper.CreateMap<Customer, CustomerCreateVM>()  
    .ForMember(f => f.Date, f => f.AddFormatter<PersianDateFormatter>());
```

و بعد واسه بر عکسش:

```
Mapper.CreateMap<CustomerCreateVM, Customer>()  
    .ForMember(f => f.Date, f => f.AddFormatter<DateTimeFormatter>());
```

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۱۲/۰۷ ۱۴:۰۶

بله میتونید.

ولی [گویا](#) برای این کار یا Flatten و Unflatten کردن; فریمورک [valueinjecter](#) توصیه میشه .

نویسنده: نوید
تاریخ: ۱۳۹۱/۱۲/۲۷ ۱۲:۰۱

ممنون از مطلب مفیدتون

من از کلاس کمکی ای که برای نگاشت چند کلاس به یک کلاس معرفی کردید استفاده کردم ولی به یک مشکل برخورد کرد.

من یک کلاس کالا دارم و یک کلاس برند که میخوام این دو رو به یک کلاس دیگه نگاشت بدم.

در اون کلاسی که معرفی کردید یک به یک کلاسهای منبع را نگاشت میدهد. هنگام استفاده وقتی که که مثلا اول کلاس کالا و بعد

کلاس برند رو برای نگاشت ارسال میکنم ، کلیه پارامترهایی که از کلاس کالا در کلاس مقصد وجود دارند مقادیر صفر و Null

میگیرند و فقط مقادیر کلاس برند در کلاس مقصد مقادیر درستشان را دارند. البته من در متد configure هنگام برای نگاشت کلاس

دوم پارامترهای کلاس اول را Ignore کردم. ولی باز هم همین مشکل پیش اومد!

یک سوال دیگه هم داشتم :

بهتر نیست به جای اینکه ما در Controller دو تابع از هر کلاس را فراخوانی کنیم(یک تابع برای دریافت لیست کالاها و یکی برای

لیست برندها) و کار نگاشت این دو به صورتیکه معرفی شد انجام دهیم، یک تابع که ترکیبی از دو کلاس را برمیگرداند (تابعی که با

استفاده از Join کلاسها مقادیر مورد نظرمان را برگرداند، یعنی برند هر کالا را به آن بچسباند) را اجرا کرده و سپس مقادیر آن را

به صورت معمول نگاشت کنیم؟

با سپاس و ببخشید که طولانی شد

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۱۲/۲۷ ۱۲:۵۳

چک کن قبل مپ کردن پراپرتی هات نال نباشن. برای بررسی دقیقتر کد رو بزار

شدنش میشه و بستگی به طراحی شما تو لایه سرویس داره و در اون صورت دیگه نیازی به استفاده از این کلاس کمکی هم

نداری.

نویسنده: نوید

تاریخ: ۱۶:۱۳ ۱۳۹۱/۱۲/۲۷

ممنون از پاسختون.

پراپرتی‌ها مقادیر درستی دارند و بعد از نگاشت مقادیر Null میگیرند.

این هم کدها :

```
public class Kala
{
    [Key]
    public int Kala_id { get; set; }

    [DisplayName("نام کالا")]
    public string Name { get; set; }

    [DisplayName("قیمت خرید")]
    public double Fee_Kharid { get; set; }

    public virtual Brand Brand { get; set; }
}

public class Brand
{
    [Key]
    public int Brand_id { get; set; }
    public string Brand_Name { get; set; }
    public virtual ICollection<Kala> Kalas { get; set; }
}

public class KalaViewModel
{
    public int Kala_Id { get; set; }
    public string Name { get; set; }
    public double Fee_Kharid { get; set; }
    public string Brand_Name { get; set; }
}

//Controller
[HttpGet]
public ActionResult Index()
{
    var kala = _Kala_Service.GetAllKalas();
    var brand = _Brand_Service.GetAllBrands();

    var kalaviewmodel = AutoMapper.Map<List<KalaViewModel>>(kala, brand);
    return View(kalaviewmodel);
}

protected override void Configure()
{
    Mapper.CreateMap<Kala, KalaViewModel>();

    Mapper.CreateMap<Brand, KalaViewModel>()
        .ForMember(des => des.Kala_Id, op => op.Ignore())
        .ForMember(des => des.Name, op => op.Ignore())
        .ForMember(x => x.Fee_Kharid, opt => opt.Ignore());
}
```

سپاس

نویسنده: علیرضا پایدار

تاریخ: ۱۱:۲۱ ۱۳۹۱/۱۲/۲۸

```
public class Kala
{
    [Key]
    public int Kala_id { get; set; }
```

```

        [DisplayName("نام کالا")]
        public string Name { get; set; }

        [DisplayName("قیمت خرید")]
        public double Fee_Kharid { get; set; }

        public virtual Brand Brand { get; set; }
    }

    public class Brand
    {
        [Key]
        public int Brand_id { get; set; }
        public string Brand_Name { get; set; }
        public virtual ICollection<Kala> Kalas { get; set; }
    }

    public class KalaViewModel
    {
        public int Kala_Id { get; set; }
        public string Name { get; set; }
        public double Fee_Kharid { get; set; }
        public string Brand_Name { get; set; }
    }

    //Controller
    [HttpGet]
    public ActionResult Index()
    {
        var kala = _Kala_Service.GetAllKalas();
        var brand = _Brand_Service.GetAllBrands();

        var kalaviewmodel = AutoMapper.Map<List<KalaViewModel>>(kala, brand);
        return View(kalaviewmodel);
    }

    protected override void Configure()
    {
        Mapper.CreateMap<Kala, KalaViewModel>()
            .ForMember(des => des.Brand_Name, op => op.MapFrom(src => src.Brand.Brand_Name));
    }

```

کد پایین از لحاظ منطقی هم درست نیست
map کردن Brand با KalaViewModel معنایی ندارد

نویسنده: محمد صاحب

تاریخ: ۱۳۹۱/۱۲/۲۸ ۱۲:۱۰

تابع ignore باعث عدم مپ کردن اون پراپرتی میشه ولی همونطور که ذکر شده AutoMapper براساس قراردادهای کار میکنه و این یه قراردادیه که پراپرتی که در مبدا معادلی براش در مقصد نباشه به صورت دیفالت ignore میشه پس نیازی به ignore نیست.

```

Mapper.CreateMap<Kala, KalaViewModel>()
    .ForMember(des => des.Brand_Name, op => op.MapFrom(src => src.Brand.Brand_Name));

var kalas = new[]
{
    new Kala
    {
        Kala_id = 1,
        Brand = new Brand {Brand_id = 1, Brand_Name = "Nike"},
        Fee_Kharid = 150000,
        Name = "Shoes"
    }, new Kala
    {
        Kala_id = 2,
        Brand = new Brand {Brand_id = 1, Brand_Name = "Nike"},
        Fee_Kharid = 12000,
        Name = "Shirt"
    }
};

```

```
var kalaviewmodel = Mapper.Map<Kala[], KalaViewModel[]>(kalas);
```

نویسنده:

نوید

تاریخ:

۱۱:۱۸ ۱۳۹۱/۱۲/۲۹

سپاس

نویسنده:

نوید

تاریخ:

۱۱:۵۹ ۱۳۹۱/۱۲/۲۹

ممنون از راهنماییتون .

یک سوال دیگه هم برام پیش اومد: اگه نحوه ارتباط کلاس ها به صورت زیر باشه:

```
public class Kala
{
    [Key]
    public int Kala_id { get; set; }

    [DisplayName("نام کالا")]
    public string Name { get; set; }

    [DisplayName("قیمت خرید")]
    public double Fee_Kharid { get; set; }

    public virtual Brand Brand { get; set; }

    public ICollection<Anbar_Kala> Anbar_Kalas { get; set; }
}

public class Anbar_Kala
{
    [ForeignKey("Anbar_Id")]
    public virtual Anbar Anbar { get; set; }
    public int Anbar_Id { get; set; }

    [ForeignKey("Kala_Id")]
    public virtual Kala Kala { get; set; }
    public int Kala_Id { get; set; }

    [DisplayName("تعداد")]
    public int Tedad { get; set; } // تعداد کالاها در هر انبار
}

public class KalaViewModel
{
    public int Kala_Id { get; set; }
    public string Name { get; set; }
    public double Fee_Kharid { get; set; }
    public string Brand_Name { get; set; }
    public int Tedad { get; set; }
}

//controller
var kala = _Kala_Service.GetAllKalas();
var tedad= _Anbar_Kala_Service.GetAllAnbar_Kalas();

var kalaviewmodel = EntityMapper.Map<List<KalaViewModel>>(kala, tedad);

protected override void Configure()
{
    Mapper.CreateMap<Kala, KalaViewModel>()
    .ForMember(des => des.Brand_Name, op => op.MapFrom(src => src.Brand.Brand_Name));

    Mapper.CreateMap<Anbar_Kala, KalaViewModel>(); // این نگاشت باید به چه صورتی باشد؟
    .ForMember(des =>des.Kala_Id, op=>op.Ignore());
}
```

تو شرایط فوق که نحوه ارتباط کلاسها به صورت عکس حالت قبله، اگر به صورت بالا نگاشت صورت بگیره، قبل از نگاشت پراپرتیها نال نیستند ولی بعد از نگاشت باز هم پارامترهای مربوط به کلاس کالا که در Modelview قرار دارند Null میشوند.

نویسنده: علیرضا پایدار
تاریخ: ۱۸:۵۷ ۱۳۹۱/۱۲/۲۹

دوست عزیز آوردن پراپرتی Tedad در کلاس KalaViewModel اشتباهه و همچنین map کردن Anbar_kala با آن. کار زیر را میتونی انجام بدی:

```
public class Anbar_KalaViewModel
{
    public Anbar Anbar { get; set; }
    public Kala Kala { get; set; }
    public int Tedad { get; set; }
}
//Class Configure
Mapper.CreateMap<Anbar_Kala,Anbar_KalaViewModel>();
```

نویسنده: حسن
تاریخ: ۱۹:۱۸ ۱۳۹۱/۱۲/۲۹

یک توصیه: مطلب ارزنده « [اصول و قراردادهای نام‌گذاری در دات‌نت](#) » بهتر است در کدهای شما رعایت شود.

نویسنده: مرتضی ریسی
تاریخ: ۱۳:۱۱ ۱۳۹۴/۰۵/۰۱

سلام.
استفاده از Ignore برای من اجباری شده. بدین صورت که مدل ویو من تعداد (نسبتا زیادی) پروپرتی نسبت به مدل برنامه ام، کم داره. حالا موقع تبدیل از مدل به ویو مدل مشکلی ندارم. ولی برعکس این موضوع خطای Unmapped members were found میده به همراه لیست این پروپرتی ها:

Unmapped properties

Salt

LastLoginDateTime

BanedDateTime

LastUpdateDateTime

تعدادی از فیلدها رو Ignore کردم و از این لیست حذف شدند.

طبق گفته شما استفاده از Ignore اختیاریه ولی برای من اجباری شده. اگه این پروپرتیها رو Ignore کنم خطایی بهم نخواهد داد. میشه راهنمایی بفرمائید تا بصورت کلی بتونم از Ignore استفاده کنم و نخوام برا تک تک فیلدها اینکارو انجام بدم.

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۴:۷ ۱۳۹۴/۰۵/۰۱

« [استفاده از Auto-ignore در AutoMapper](#) »