

در سناریوهای متداول نگاشت اشیاء، مشخص است که نوع ViewModel برنامه چیست و معادل Model آن کدام است. اما حالت‌هایی مانند کار با anonymous objects و یا data reader و data table و امثال آن نیز وجود دارند که در این حالت‌ها، نوع منبع داده‌ی مورد استفاده، شیء مشخصی نیست که بتوان آن‌را در قسمت CreateMap مشخص کرد. برای مدیریت یک چنین حالت‌هایی، متد DynamicMap طراحی شده‌است.

### مثال اول: تبدیل یک DataTable به لیست جنریک معادل

فرض کنید یک DataTable را با ساختار و داده‌های ذیل در اختیار داریم:

```
var dataTable = new DataTable("SalaryList");
dataTable.Columns.Add("User", typeof (string));
dataTable.Columns.Add("Month", typeof (int));
dataTable.Columns.Add("Salary", typeof (decimal));

var rnd = new Random();
for (var i = 0; i < 200; i++)
    dataTable.Rows.Add("User " + i, rnd.Next(1, 12), rnd.Next(400, 2000));
```

نوع این DataTable کاملاً پویا است و می‌تواند هر بار در قسمت‌های مختلف برنامه تعریف متفاوتی داشته باشد. در ادامه معادل کلاس ساختار ستون‌های این DataTable را به صورت ذیل تهیه می‌کنیم.

```
public class SalaryList
{
    public string User { set; get; }
    public int Month { set; get; }
    public decimal Salary { set; get; }
}
```

اکنون می‌خواهیم اطلاعات DataTable را به لیستی جنریک از SalaryList نگاشت کنیم. برای اینکار تنها کافی است از متد DaynamicMap استفاده نماییم:

```
var salaryList = AutoMapper.Mapper.DynamicMap<IDataReader,
List<SalaryList>>(dataTable.CreateDataReader());
```

منبع داده را از نوع IDataReader بر اساس متد CreateDataReader مشخص کرده‌ایم. به این ترتیب AutoMapper قادر خواهد بود تا اطلاعات این DataTable را به صورت خودکار پیمایش کند. سپس مقصد را نیز لیست جنریکی از کلاس SalaryList تعیین کرده‌ایم. مابقی کار را متد DynamicMap انجام می‌دهد.

کار با AutoMapper نسبت به [راه حل‌های Reflection متداول](#) بسیار سریعتر است. زیرا AutoMapper از مباحث [Fast reflection](#) به صورت توکار استفاده می‌کند.

### مثال دوم: تبدیل لیستی از اشیاء anonymous به لیستی جنریک

در اینجا قصد داریم یک شیء anonymous را به شیء معادل SalaryList آن نگاشت کنیم. این کار را نیز می‌توان توسط متد DynamicMap انجام داد:

```
var anonymousObject = new
{
    User = "User 1",
    Month = 1,
    Salary = 100000
}
```

```
};  
var salary = Mapper.DynamicMap<SalaryList>(anonymousObject);
```

و یا نمونه‌ی دیگر آن تبدیل یک لیست anonymous به معادل جنریک آن است که به نحو ذیل قابل انجام است:

```
var anonymousList = new[]  
{  
    new  
    {  
        User = "User 1",  
        Month = 1,  
        Salary = 100000  
    },  
    new  
    {  
        User = "User 2",  
        Month = 1,  
        Salary = 300000  
    }  
};  
var salaryList = anonymousList.Select(item => Mapper.DynamicMap<SalaryList>(item)).ToList();
```

این نکته در مورد حاصل کوئری‌های LINQ یا IQueryable ها نیز صادق است.

### مثال سوم: نگاشت پویا به یک اینترفیس

فرض کنید یک چنین اینترفیسی، در برنامه تعریف شده‌است و همچنین دارای هیچ نوع پیاده سازی هم در برنامه نیست:

```
public interface ICustomerService  
{  
    string Code { get; set; }  
    string Name { get; set; }  
}
```

اکنون قصد داریم یک شیء anonymous را به آن نگاشت کنیم:

```
var anonymousObject = new  
{  
    Code = "111",  
    Name = "Test 1"  
};  
var result = Mapper.DynamicMap<ICustomerService>(anonymousObject);
```

در این حالت خاص، AutoMapper با استفاده از یک [Dynamic Proxy](#) به نام LinFu (که با اسمبلی آن Merge شده‌است)، پیاده سازی پویایی را از اینترفیس مشخص شده تهیه کرده و سپس کار نگاشت را انجام می‌دهد.

کدهای کامل این قسمت را از اینجا می‌توانید دریافت کنید:

[AM\\_Sample05.zip](#)