عنوان: **استفاده از AutoMapper در برنامه‌های چند ریسمانی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۲۵ ۲۰:۳۹
آدرس: www.dotnettips.info
گروه‌ها: MVC, Entity framework, AutoMapper

نکته‌ی بسیار مهمی را که حین کار با AutoMapper باید بخاطر داشت، عدم thread safety متد **CreateMap**. Mapper آن است و استفاده‌ی از آن در برنامه‌های چند ریسمانی و خصوصا برنامه‌های وب، مشکلات متعددی را به همراه خواهد داشت. بنابراین بهترین محل تعریف و معرفی این نگاشت‌ها، در حین آغاز برنامه‌است؛ برای مثال در متد Application_Start فایل global.asax برنامه‌های وب، یا ابتدای متد Main برنامه‌های دسکتاپ.
برای نمونه یک چنین کدی را نباید در برنامه‌های خود داشته باشید:

```
public ActionResult Index()
{
    Mapper.CreateMap<UserViewModel, User>();
    //ادامه‌ی کدها
```

در اینجا از متد استاتیک **CreateMap**. Mapper ، در یک اکشن متد برنامه‌ی ASP.NET MVC استفاده شده‌است. این متد thread safe نیست و چون کار تنظیمات اولیه‌ی این نگاشت‌ها (پیش از کش شدن آن‌ها) اندکی زمانبر است، ممکن است در این بین، دو کاربر همزمان به این قطعه کد رسیده و شاهد این باشند که تعدادی از خواص در اینجا نگاشت نشده‌اند.

نمونه‌ی دیگر آن، یک چنین کدهایی هستند:

```
using (var context = new TestDbContext())
    {
        Mapper.CreateMap<SourceClass, DestinationClass>()
            .AfterMap((src, dest) =>
            {
                  //using context
            });

         var dest = Mapper.Map<DestinationClass>(source);
    }
```

در اینجا برحسب نیاز از context مربوط به Entity framework داخل تنظیمات Mapper.CreateMap استفاده شده‌است. متد **CreateMap**. Mapper استاتیک است و context استفاده شده‌ی در آن thread safe نیست. همینجا است که مشکلات تخریب اطلاعات را شاهد خواهید بود.
اگر در یک چنین حالتی نیاز به استفاده‌ی context داشتید، بهتر است متدهای استاتیک AutoMapper را فراموش کرده و به نحو ذیل یک موتور محلی نگاشت را ایجاد کنید. چون سطح دید و دسترسی این موتور، عمومی و سراسری نیست، مشکلات thread safety را نخواهد داشت.

```
var configurationStore = new ConfigurationStore(new TypeMapFactory(), MapperRegistry.Mappers);
configurationStore.AddProfile<TestProfile1>();
var mapper = new MappingEngine(configurationStore);
configurationStore.CreateMap<SourceClass, DestinationClass>()
//ادامه‌ی کدها
```