



auto^{Mapper}mapper

GEÇİŞ



Mapster

Neden

.NET geliřtiricileri iin byk bir deęiřim! AutoMapper, Temmuz 2025 itibariyle tamamen ticari bir rn haline geldi. Yıllarca cretsiz kullandığımız bu ktphane artık cretli lisans modeline geti.

Community License: Yıllık geliri 5M\$ altındaki řirketler iin cretsiz
Commercial License: Takım byklęne gre fiyatlandırma


```

2 references
public class Customer
{
    1 reference
    public string Name { get; set; } = string.Empty;
    1 reference
    public string Surname { get; set; } = string.Empty;
    1 reference
    public string Password { get; set; } = string.Empty;
}

2 references
public class CustomerDTO
{
    1 reference
    public string Name { get; set; } = string.Empty;
    1 reference
    public string Surname { get; set; } = string.Empty;

    1 reference
    public override string ToString()
    {
        return $"Name = {Name}\nSurname = {Surname}";
    }
}

```

```

Customer customer = new Customer()
{
    Name = "Ahmet Taha", Surname = "Mücasiroğlu", Password = "123456"
};
CustomerDTO customerDTO = mapper.Map<CustomerDTO>(customer);
Console.WriteLine(customerDTO.ToString());
/*Çıktı
    Name = Ahmet Taha
    Surname = Mücasiroğlu
*/

```

Elimizde 1 adet entity ve 1 adet dto sınıfımız olsun. Automapper kullanarak dönüşümlerde, hiç bir ek özellik olmasa bile, `CrateMap<Kaynak, Hedef>()`; şeklinde bir tanımlama yapılmalıydı. Mapster ise bunu otomatik yapmakta.

Eğer dönüşüm sırasında ek işlemler olacak ise sizden Profile mantığında bir `IRegister` sınıfı istemektedir. Bu özelliği sayesinde Automapper'a göre daha hızlı kurulum ve kullanım sağlamakta.

@csharp_gunlukleri

3 references

```
public class UserEntity
```

```
{
```

1 reference

```
public string Name { get; set; } = string.Empty;
```

1 reference

```
public string Surname { get; set; } = string.Empty;
```

1 reference

```
public string Password { get; set; } = string.Empty;
```

2 references

```
public DateTime Birthdate { get; set; }
```

```
}
```

3 references

```
public class UserDTO
```

```
{
```

1 reference

```
public string Name { get; set; } = string.Empty;
```

1 reference

```
public string Surname { get; set; } = string.Empty;
```

1 reference

```
public DateTime Birthdate { get; set; }
```

2 references

```
public bool IsBirthdate { get; set; }
```

1 reference

```
public override string ToString()
```

```
{
```

```
return $"Name = {Name}" +
```

```
    $"\\nSurname = {Surname}" +
```

```
    $"\\nBirthdate = {Birthdate}" +
```

```
    $"\\nIsBirthdate = {IsBirthdate}";
```

```
}
```

```
}
```

0 references

```
public class UserMapProfile : IRegister
```

```
{
```

0 references

```
public void Register(TypeAdapterConfig config)
```

```
{
```

```
    config.NewConfig<UserEntity, UserDTO>()
```

```
        .Map(dst => dst.IsBirthdate, src => src.Birthdate.Date == DateTime.Now.Date);
```

```
}
```

```
}
```

Burada ise bir adet User tablomuz var. Sadece örnek olması için eğer doğum günü bugün ise sadece DTO sınıfında olan IsBirthdate değeri true olsun demek istersek dönüşümü UserMapProfile sınıfı içinde yönetebiliriz. Burada tüm configleri tek sınıfta ekleyebilir yada sınıf sınıf ayırarak daha düzenli yapabiliriz.


```

HostApplicationBuilder builder = Host.CreateApplicationBuilder();

// api yada mvc projesi gibi servis ekleme kısmı
builder.Services.AddMapster();

// projeyi tara ve IRegsiter ayarlamalarını bul
TypeAdapterConfig.GlobalSettings.Scan(Assembly.GetExecutingAssembly());

IHost app = builder.Build();

//hazırlanmış mapper sınıfını alma
 IMapper mapper = app.Services.GetRequiredService<IMapper>();

UserEntity entity = new UserEntity()
{
    Name = "Ahmet Taha",
    Surname = "Mücasiroğlu",
    Password = "123456",
    Birthdate = DateTime.Now
};

UserDTO dto = mapper.Map<UserDTO>(entity);
Console.WriteLine(dto.ToString());

/* Çıktı
    Name = Ahmet Taha
    Surname = Mücasiroğlu
    Birthdate = 16.08.2025 14:53:43
    IsBirthdate = True
*/

```

Burada dönüşüm için özel ayarı Assembly kullanarak otomatik bulmasını sağlarız. Sonrasında Config eklenmiş Mapster sınıfını kullanarak dönüşümü başarı ile tamamlarız.

Automapper'a fark attığı bir konuda burası, eğer config olmazsa veya bulamazsa hata vermez. Birthdate false olur. Evet veri hatası olmakta bu sefer ama en azından program çalışmaya devam eder. İki tarafından kendine özgü yönleri bulunmaktadır.