



TOPIC: AUTO MAPPING IN SOFTWARE DEVELOPMENT

September 2016

V_3_REV-8_04_16

AGENDA



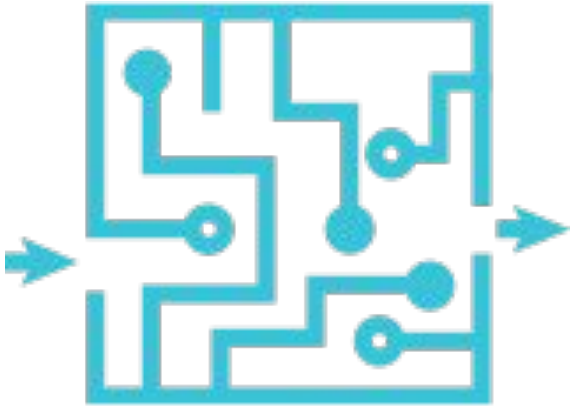
#1 Formulation of the problem

#2 Using AutoMapper in .NET

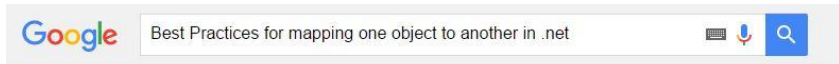
#3 Using TinyMapper in .NET

#1 Formulation of the problem

How to map one object (objects) on another object?



WHAT GOOGLE SAID?



Все Видео Картинки Новости Ещё ▾ Инструменты поиска

Результатов: примерно 1 260 000 (0,76 сек.)

c# - Best Practices for mapping one object to another - Stack Overflow
stackoverflow.com/.../best-practices-for-mapping-one-object... ▾ Перевести эту страницу
20 апр. 2013 г. - I would opt for AutoMapper, an open source and free mapping library which allows to map one type into another, based on conventions (i.e. map ...
Вы посетили эту страницу несколько раз (2). Дата последнего посещения: 18.09.16

.net - Best Practices For Mapping DTO to Domain Object? - Stack ...
stackoverflow.com/.../best-practices-for-mapping-dto-to-do... ▾ Перевести эту страницу
24 мар. 2009 г. - A benefit of having a mapper that sits between your domain and your DTO is not as ... Another idea put forth is for the Domain Object to "contain" the DTO, ... You could use an automapper such as the one written by Jimmy ...

c# - Object to object mapper - Stack Overflow
stackoverflow.com/questions/.../object-to-object-mapper ▾ Перевести эту страницу
13 нояб. 2008 г. - You are looking for AutoMapper. http://automapper.codeplex.com/ ... An example is I have a two web services, one has Investors another has ... And here you can find good example on AutoMapper: codearsenal.net/2012/12/... Movies & TV · Music: Practice & Theory · Seasoned Advice (cooking) · Home ...

c# - Apply properties values from one object to another of the same ...
stackoverflow.com/.../apply-properties-values-from-one-obj... ▾ Перевести эту страницу
30 мая 2009 г. - I have a type in MiscUtil called PropertyCopy which does something similar GetType(); // Collect all the valid properties to map var results = from srcProp in typeSrc. Best practices to manage entities while querying the data out of tables? 0 - How to copy properties from one .net object to another.

c# - Copy values from one object to another - Stack Overflow
stackoverflow.com/.../copy-values-from-one-object-to-anoth... ▾ Перевести эту страницу
12 апр. 2010 г. - Jon Skeet and Marc Gravell have a library called MiscUtil. Inside MiscUtil.Reflection ... a suggestion for a good utility class that maps values from one object to another? ... NET 3.5. It works by running over the public properties of the Movies & TV · Music: Practice & Theory · Seasoned Advice (cooking) ...

10 Answers

active oldest votes



You are looking for AutoMapper

23

<http://automapper.codeplex.com/>



share improve this answer



answered Apr 11 '09 at 13:51



Matt Hinze

11.3k ● 2 ● 24 ● 34

USING automapper IN .NET

```
public class UserDto
{
    public string FirstName { get; set; }

    public string LastName { get; set; }

    public string Email { get; set; }
}

public class User
{
    public string FullName { get; set; }

    public string Email { get; set; }
}
```

```
Install-Package AutoMapper
```

```
User user = Mapper.Map<User>(new UserDto
{
    FirstName = "Ivan",
    LastName = "Ivanov",
    Email = "ivan@ivanon.ru"
});
```

```
Mapper.CreateMap<UserDto, User>()
    .ForMember(x => x.FullName,
        x => x.MapFrom(m => m.FirstName + " " + m.LastName))
    .ForMember(x => x.Email,
        x => x.MapFrom(m => m.Email));
```

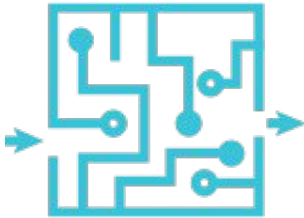
USING AUTO MAPPER IN .NET - Conditional Object Mapper

Conditional Object Mappers make new type maps based on conditional between the source and the destination type.

```
var config = new MapperConfiguration(cfg => {  
    cfg.AddConditionalObjectMapper().Where((s, d) => s.Name == d.Name + "Dto");  
});
```

USING AUTO MAPPER IN .NET - Conditional Object Mapper

```
2 using System;
3
4 namespace AutoMapperTestProject.Classes
5 {
6     4 references
7     public class FromClass
8     {
9         0 references
10        public FromClass() { }
11
12        1 reference
13        public FromClass(
14            string firstName,
15            string lastName,
16            string someField,
17            bool isSomeValue,
18            DateTime someDate)
19        {
20            FirstName = firstName;
21            LastName = lastName;
22            SomeField = someField;
23            IsSomeValue = isSomeValue;
24            DOB = someDate;
25        }
26
27        2 references
28        public string FirstName { get; protected set; }
29
30        2 references
31        public string LastName { get; protected set; }
32
33        1 reference
34        public string SomeField { get; protected set; }
35
36        2 references
37        public bool IsSomeValue { get; protected set; }
38
39        6 references
40        public DateTime DOB { get; protected set; }
41    }
42 }
```



```
namespace AutoMapperTestProject.Classes
{
    0 references
    public class ToClassDto
    {
        0 references
        public string FullName { get; set; }
        0 references
        public int YearOfBirth { get; set; }
        0 references
        public int MonthOfBirth { get; set; }
        0 references
        public int DayOfBirth { get; set; }
        0 references
        public string Notation { get; set; }
        0 references
        public int? Age { get; set; }
    }
}
```

USING AUTO MAPPER IN .NET - Conditional Object Mapper

```
file:///C:/Users/Serhii_Voznyi/Desktop/AutoMapperTestProject/AutoMapperTestProject/bin/Debug/AutoMapperTestProject.EXE

 AutoMapperTestProject.Classes.FromClass
 FirstName= 'First'
 LastName= 'Last'
 SomeField= 'Some guid!'
 IsSomeValue= 'True'
 DOB= '9/20/1996 7:47:20 AM'

 AutoMapperTestProject.Classes.ToClass
 FullName= 'First Last'
 YearOfBirth= '1996'
 MonthOfBirth= '9'
 DayOfBirth= '20'
 Notation= 'This field meets all conditions!'
 Age= '20'

)
.ForMember(
    x => x.Notation,
    x => x.MapFrom(m => "This field meets all conditions!"))
.ForMember(
    x => x.Notation,
    x => x.Condition(
        src => src.DOB <= DateTime.UtcNow.AddYears(-18)|
        && src.IsSomeValue))
;
```


USING AUTO MAPPER IN .NET - CUSTOM TYPE CONVERTERS

```
Mapper.Initialize(cfg => {  
    cfg.CreateMap<string, int>().ConvertUsing(Convert.ToInt32);  
    cfg.CreateMap<string, DateTime>().ConvertUsing(new DateTimeTypeConverter());  
    cfg.CreateMap<string, Type>().ConvertUsing<TypeTypeConverter>();  
    cfg.CreateMap<Source, Destination>();  
});
```

USING AUTO MAPPER IN .NET



I would opt for [AutoMapper](#), an open source and free mapping library which allows to map one type into another, based on conventions (i.e. map public properties with the same names and same/derived/convertible types, along with many other [smart ones](#)). Very easy to use, will let you achieve something like this:

```
Model model = Mapper.Map<Model>(dto);
```

Not sure about your specific requirements, but AutoMapper also supports [custom value resolvers](#), which should help you writing a single, generic implementation of your particular mapper.

share improve this answer

answered Apr 20 '13 at 10:02



[Efran Cobisi](#)

3,820 ● 10 ● 16

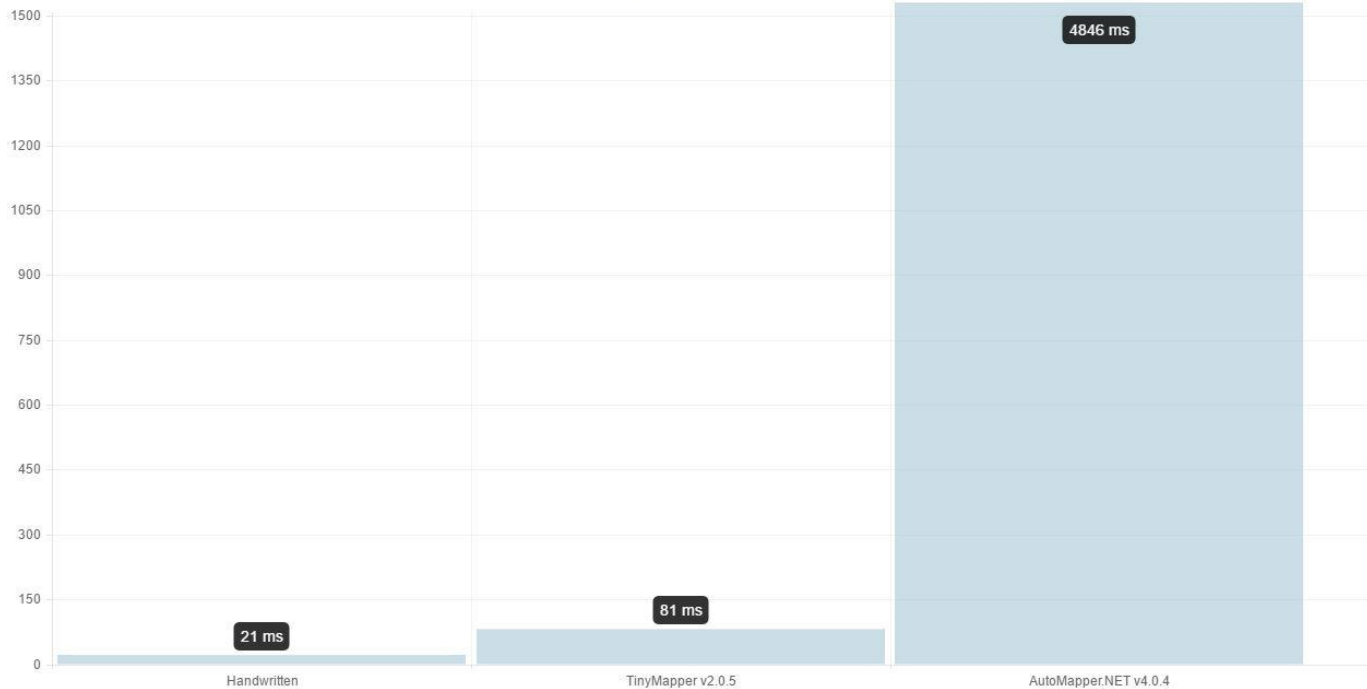
-
- 1 We have used automapper before, but we have sense dropped it due to it's slow performance. – [Alex](#) Apr 22 '13 at 15:02
 - 1 Agreed. Done the same thing, ended up with more work dropping automapper then writing a custom one in the first place. Automapper has very very slow performance – [ZolaKt](#) Jun 10 '15 at 12:27
-

USING TINY MAPPER IN .NET



USING TINY MAPPER IN .NET

Speed comparison results
(Lower is better)



USING TINY MAPPER IN .NET

```
TinyMapper.Bind(s => s.FullName, d => d.FullName);  
config.Bind(s => s.FullName, d => d.FullName);  
config.Bind(s => s.FullName, d => d.FullName);  
config.Bind(s => s.FullName, d => d.FullName);  
config.Bind(s => Date, d => d.Age);  
config.Bind(s => s.FullName, d => d.FullName);  
    s =>  
        s.IsSomeValue  
        s.DOB <= s.DOB  
        ?  
    d =>
```

USING TINY MAPPER IN .NET

```
1 reference
public sealed class FromClassConverter : TypeConverter
{
    0 references
    public override bool CanConvertTo(ITypeDescriptorContext context, Type destinationType)
    {
        return destinationType == typeof(ToClass);
    }

    0 references
    public override object ConvertTo(ITypeDescriptorContext context, CultureInfo culture, object value, Type destinationType)
    {
        var concreteValue = (FromClass)value;
        var result = new ToClass(
            string.Format("{0} {1}", concreteValue.FirstName, concreteValue.LastName),
            concreteValue.DOB.Year,
            concreteValue.DOB.Month,
            concreteValue.DOB.Day,
            concreteValue.IsSomeValue &&
            concreteValue.DOB <= DateTime.UtcNow.AddYears(-18)
                ? "This field meets all conditions!"
                : string.Empty,
            DateTime.UtcNow.Year - concreteValue.DOB.Year);
        return result;
    }
}
```



THANK YOU FOR WATCHING



Subscribe

WE ARE



epam