



Kostenloses eBook

LERNEN

automapper

Free unaffiliated eBook created from
Stack Overflow contributors.

#automapp

er

Inhaltsverzeichnis

Über	1
Kapitel 1: Erste Schritte mit automapper	2
Bemerkungen.....	2
Examples.....	2
Installation oder Setup.....	2
Kapitel 2: Profile	3
Syntax.....	3
Parameter.....	3
Examples.....	3
Grundprofil.....	3
Laden aller Profile in einer Baugruppe.....	4
Credits	6



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [automapper](#)

It is an unofficial and free automapper ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official automapper.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit automapper

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was ein Automapper ist und warum ein Entwickler ihn verwenden möchte.

Es sollte auch alle großen Themen in automapper erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für Automapper neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Examples

Installation oder Setup

Automapper kann von [nuget](#) installiert werden, [indem](#) der folgende Befehl in der Package Manager Console ausgeführt wird:

```
Install-Package AutoMapper
```

Dann kann es `using` Anweisungen in verschiedene Dateien des Projekts eingefügt werden, in dem es installiert wurde:

```
using AutoMapper;
```

[Erste Schritte mit automapper online lesen: https://riptutorial.com/de/automapper/topic/5254/erste-schritte-mit-automapper](https://riptutorial.com/de/automapper/topic/5254/erste-schritte-mit-automapper)

Kapitel 2: Profile

Syntax

- `public void AddProfiles(params string[] assemblyNamesToScan)`
- `public void AddProfiles(params Assembly[] assembliesToScan)`
- `public void AddProfiles(params Type[] typesFromAssembliesContainingProfiles)`
- `public void AddProfiles(IEnumerable<string> assemblyNamesToScan)`
- `public void AddProfiles(IEnumerable<Assembly> assembliesToScan)`
- `public void AddProfiles(IEnumerable<Type> typesFromAssembliesContainingProfiles)`

Parameter

Parameter	Einzelheiten
<code>assemblyNamesToScan</code>	Assemblynamen mit Profilen zum Laden und Scannen.
<code>assembliesToScan</code>	Baugruppen, die zu scannende Profile enthalten.
<code>typesFromAssembliesContainingProfiles</code>	Typen aus Baugruppen mit Profilen zum Laden und Scannen

Examples

Grundprofil

Profile erlauben dem Programmierer, Karten in Klassen zu organisieren, wodurch die Lesbarkeit und Wartbarkeit des Codes verbessert wird. Es können beliebig viele Profile erstellt und bei Bedarf zu einer oder mehreren Konfigurationen hinzugefügt werden. Profile können sowohl mit statischen als auch mit instanzbasierten APIs verwendet werden.

```
public class User
{
    public int Id { get; set; }
    public string Username { get; set; }
    public string Password { get; set; }
    public string DisplayName { get; set; }
    public string Email { get; set; }
    public string PhoneNumber { get; set; }
}

public class UserViewModel
{
    public string DisplayName { get; set; }
    public string Email { get; set; }
}
```

```

public class MappingProfile : Profile
{
    public MappingProfile()
    {
        CreateMap<User, UserViewModel>();
    }
}

public class Program
{
    static void Main(string[] args)
    {
        Mapper.Initialize(cfg => {
            cfg.AddProfile<MappingProfile>();
            //cfg.AddProfile(new MappingProfile()); // Equivalent to the above
        });

        var user = new User()
        {
            Id = 1,
            Username = "jdoe",
            Password = "password",
            DisplayName = "John Doe",
            Email = "jdoe@example.com",
            PhoneNumber = "555-123-4567"
        };

        var userVM = Mapper.Map<UserViewModel>(user);

        Console.WriteLine("DisplayName: {0}\nEmail: {1}", userVM.DisplayName, userVM.Email);
    }
}

```

Laden aller Profile in einer Baugruppe

Oft ist es sinnvoll, alle Profile einer oder mehrerer Baugruppen in eine Konfiguration laden zu können. AutoMapper stellt die Methode `AddProfiles` zur Verfügung, die mehrere Überladungen enthält, die das Laden von Profilen ermöglichen, indem die Assembly übergeben wird, der Assemblyname angegeben wird oder ein in der Assembly enthaltener Typ angegeben wird. Nur Klassen, die von **AutoMapper.Profile** erben, werden gefunden und der Konfiguration hinzugefügt.

Laden Sie alle Profile in einer Baugruppe, indem Sie den Namen der Baugruppe angeben:

```

Mapper.Initialize(cfg => {
    cfg.AddProfiles("MyApplication.Core", "MyApplication.Web");
});

```

Laden Sie alle Profile in einer Baugruppe, indem Sie einen Typ aus der Baugruppe angeben:

```

Mapper.Initialize(cfg => {
    cfg.AddProfiles(typeof(Student), typeof(Course));
});

```

Laden Sie alle Profile in der aktuellen Baugruppe:

```
Mapper.Initialize(cfg => {  
    cfg.AddProfiles(Assembly.GetExecutingAssembly());  
});
```

Profile online lesen: <https://riptutorial.com/de/automapper/topic/5729/profile>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit automapper	Community , meJustAndrew
2	Profile	Brett Wolfington , Tom