

 eBook Gratuit

APPRENEZ automapper

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#automapp

er

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec automapper.....	2
Remarques.....	2
Exemples.....	2
Installation ou configuration.....	2
Chapitre 2: Les profils.....	3
Syntaxe.....	3
Paramètres.....	3
Exemples.....	3
Profil de base.....	3
Chargement de tous les profils dans un assemblage.....	4
Crédits.....	6

À propos

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

Chapitre 1: Démarrer avec automapper

Remarques

Cette section fournit une vue d'ensemble de ce qu'est automapper et pourquoi un développeur peut vouloir l'utiliser.

Il devrait également mentionner tous les grands sujets dans automapper, et établir un lien avec les sujets connexes. La documentation pour automapper étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Exemples

Installation ou configuration

Automapper peut être installé à partir de [nuget en](#) exécutant la commande suivante dans Package Manager Console:

```
Install-Package AutoMapper
```

Ensuite, il peut être inclus dans différents fichiers du projet dans lequel il a été installé en `using` directives:

```
using AutoMapper;
```

Lire [Démarrer avec automapper en ligne](#):

<https://riptutorial.com/fr/automapper/topic/5254/demarrer-avec-automapper>

Chapitre 2: Les profils

Syntaxe

- `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)`

Paramètres

Paramètre	Détails
<code>assemblyNamesToScan</code>	Noms d'assemblage contenant des profils à charger et à analyser.
<code>assemblagesToScan</code>	Assemblages contenant des profils à analyser.
<code>typesFromAssembliesContainingProfiles</code>	Types d'assemblages contenant des profils à charger et à analyser

Exemples

Profil de base

Les profils permettent au programmeur d'organiser les cartes en classes, améliorant la lisibilité et la maintenabilité du code. Vous pouvez créer n'importe quel nombre de profils et les ajouter à une ou plusieurs configurations si nécessaire. Les profils peuvent être utilisés avec les API statiques et basées sur des instances.

```
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);
    }
}

```

Chargement de tous les profils dans un assemblage

Il est souvent utile de pouvoir charger tous les profils d'un ou de plusieurs assemblages dans une configuration. AutoMapper fournit la méthode `AddProfiles`, qui comporte plusieurs surcharges qui permettent de charger les profils en transmettant l'assembly, en spécifiant le nom de l'assembly ou en spécifiant un type contenu dans l'assembly. Seules les classes héritant d'**AutoMapper.Profile** seront localisées et ajoutées à la configuration.

Chargez tous les profils dans un assembly en spécifiant le nom de l'assembly:

```

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

```

Chargez tous les profils dans un assembly en spécifiant un type de l'assembly:

```

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

```

Chargez tous les profils dans l'assembly en cours:

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

Lire Les profils en ligne: <https://riptutorial.com/fr/automapper/topic/5729/les-profils>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec automapper	Community , meJustAndrew
2	Les profils	Brett Wolfington , Tom