



**EBook Gratuito**

# APPENDIMENTO automapper

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#automapp

er

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con l'automapper.....</b>	<b>2</b>
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
<b>Capitolo 2: Profili.....</b>	<b>3</b>
Sintassi.....	3
Parametri.....	3
Examples.....	3
Profilo di base.....	3
Caricamento di tutti i profili in un assieme.....	4
<b>Titoli di coda.....</b>	<b>6</b>

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](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con l'automapper

## Osservazioni

Questa sezione fornisce una panoramica di ciò che è automapper e perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare qualsiasi argomento di grandi dimensioni all'interno di automapper e collegarsi agli argomenti correlati. Poiché la documentazione di automapper è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

## Examples

### Installazione o configurazione

Automapper può essere installato da [nuget](#) con il seguente comando nella Console di Gestione pacchetti:

```
Install-Package AutoMapper
```

Quindi può essere incluso in diversi file all'interno del progetto a cui è stato installato `using` direttive:

```
using AutoMapper;
```

Leggi Iniziare con l'automapper online: <https://riptutorial.com/it/automapper/topic/5254/iniziare-con-l-automapper>

# Capitolo 2: Profili

## Sintassi

- `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)`

## Parametri

Parametro	Dettagli
<code>assemblyNamesToScan</code>	Nomi assembly contenenti profili da caricare e scansionare.
<code>assembliesToScan</code>	Assemblee contenenti profili da analizzare.
<code>typesFromAssembliesContainingProfiles</code>	Tipi da assiemi contenenti profili da caricare e scansionare

## Examples

### Profilo di base

I profili consentono al programmatore di organizzare le mappe in classi, migliorando la leggibilità del codice e la manutenibilità. È possibile creare qualsiasi numero di profili e aggiungerli a una o più configurazioni in base alle esigenze. I profili possono essere utilizzati sia con le API statiche che con quelle basate su istanze.

```
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);
    }
}

```

## Caricamento di tutti i profili in un assieme

Spesso è utile poter caricare tutti i profili in uno o più assieme in una configurazione. AutoMapper fornisce il metodo `AddProfiles`, che ha diversi overload che consentono il caricamento dei profili passando l'assembly, specificando il nome dell'assembly o specificando un tipo contenuto nell'assembly. Verranno individuate e aggiunte alla configurazione solo le classi ereditate da **AutoMapper.Profile**.

Carica tutti i profili in un assembly specificando il nome dell'assieme:

```

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

```

Carica tutti i profili in un assembly specificando un tipo dall'assembly:

```

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

```

Carica tutti i profili nell'assieme corrente:

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

Leggi Profili online: <https://riptutorial.com/it/automapper/topic/5729/profili>

---

# Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con l'automapper	<a href="#">Community</a> , <a href="#">meJustAndrew</a>
2	Profili	<a href="#">Brett Wolfington</a> , <a href="#">Tom</a>