



EBook Gratis

APRENDIZAJE automapper

Free unaffiliated eBook created from
Stack Overflow contributors.

#automapp

er

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con automapper.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Capítulo 2: Perfiles.....	3
Sintaxis.....	3
Parámetros.....	3
Examples.....	3
Perfil basico.....	3
Cargando todos los perfiles en un montaje.....	4
Creditos.....	6

Acerca de

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

Capítulo 1: Empezando con automapper

Observaciones

Esta sección proporciona una descripción general de qué es automapper y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de automapper y vincular a los temas relacionados. Dado que la Documentación para automapper es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Automapper se puede instalar desde [nuget](#) ejecutando el siguiente comando en la Consola del Administrador de Paquetes:

```
Install-Package AutoMapper
```

Luego se puede incluir en diferentes archivos dentro del proyecto en el que se ha instalado `using` directivas:

```
using AutoMapper;
```

Lea [Empezando con automapper en línea](#):

<https://riptutorial.com/es/automapper/topic/5254/empezando-con-automapper>

Capítulo 2: Perfiles

Sintaxis

- `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)`

Parámetros

Parámetro	Detalles
<code>assemblyNamesToScan</code>	Nombres de ensamblajes que contienen perfiles para cargar y escanear.
<code>asambleasToScan</code>	Asambleas que contienen perfiles para escanear.
<code>typesFromAssembliesContainingProfiles</code>	Tipos de ensamblajes que contienen perfiles para cargar y escanear

Examples

Perfil basico

Los perfiles le permiten al programador organizar mapas en clases, mejorando la legibilidad y la capacidad de mantenimiento del código. Se puede crear cualquier número de perfiles y agregarlos a una o más configuraciones según sea necesario. Los perfiles se pueden usar tanto con las API estáticas como con las basadas en instancias.

```
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);
    }
}

```

Cargando todos los perfiles en un montaje

A menudo es útil poder cargar todos los perfiles en uno o más conjuntos en una configuración. AutoMapper proporciona el método `Método AddProfiles`, que tiene varias sobrecargas que permiten que los perfiles se carguen al pasar el Ensamblado, especificando el nombre del ensamblaje o especificando un tipo contenido en el ensamblaje. Sólo las clases que se heredan de **AutoMapper.Profile** se **ubicarán** y agregarán a la configuración.

Cargue todos los perfiles en un conjunto especificando el nombre del conjunto:

```

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

```

Cargue todos los perfiles en un conjunto especificando un tipo del conjunto:

```

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

```

Cargar todos los perfiles en el montaje actual:

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

Lea Perfiles en línea: <https://riptutorial.com/es/automapper/topic/5729/perfiles>

Creditos

S. No	Capítulos	Contributors
1	Empezando con automapper	Community , meJustAndrew
2	Perfiles	Brett Wolfington , Tom