



Бесплатная электронная книга

УЧУСЬ

automapper

Free unaffiliated eBook created from
Stack Overflow contributors.

#automapp

er

.....	1
1: automapper	2
.....	2
Examples.....	2
.....	2
2:	3
.....	3
.....	3
Examples.....	3
.....	3
.....	4
.....	6

Около

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

глава 1: Начало работы с automapper

замечания

В этом разделе представлен обзор того, что такое automapper, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в automapper и ссылки на связанные темы. Поскольку Documentation for automapper является новым, вам может потребоваться создать начальные версии этих связанных тем.

Examples

Установка или настройка

Automapper можно установить из [nuget](#), выполнив следующую команду в консоли диспетчера пакетов:

```
Install-Package AutoMapper
```

Затем он может быть включен в разные файлы в рамках проекта, в который он был установлен, с `using` директив:

```
using AutoMapper;
```

Прочитайте [Начало работы с automapper онлайн](#):

<https://riptutorial.com/ru/automapper/topic/5254/начало-работы-с-automapper>

глава 2: профили

Синтаксис

- `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)`

параметры

параметр	подробности
<code>assemblyNamesToScan</code>	Названия сборок, содержащие профили для загрузки и сканирования.
<code>assembliesToScan</code>	Ассембли, содержащие профили для сканирования.
<code>typesFromAssembliesContainingProfiles</code>	Типы сборок, содержащих профили для загрузки и сканирования

Examples

Основной профиль

Профили позволяют программисту организовывать карты в классы, улучшая читаемость кода и поддерживаемость. Любое количество профилей может быть создано и добавлено в одну или несколько конфигураций по мере необходимости. Профили могут использоваться как с статическими, так и с интерфейсами на основе экземпляров.

```
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);
    }
}

```

Загрузка всех профилей в сборку

Часто полезно иметь возможность загружать все профили в одну или несколько сборок в конфигурацию. AutoMapper предоставляет метод `AddProfiles` метода, который имеет несколько перегрузок, которые позволяют загружать профили, передавая сборку, указывая имя сборки или указывая тип, содержащийся в сборке. Только классы, наследующие от **AutoMapper.Profile**, будут расположены и добавлены в конфигурацию.

Загрузите все профили в сборку, указав имя сборки:

```

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

```

Загрузите все профили в сборку, указав тип из сборки:

```

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

```

Загрузите все профили в текущую сборку:

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

Прочитайте профили онлайн: <https://riptutorial.com/ru/automapper/topic/5729/профили>

кредиты

S. No	Главы	Contributors
1	Начало работы с automapper	Community , meJustAndrew
2	профили	Brett Wolfington , Tom