# LEARNING automapper

#automapper

# Table of Contents

# About

# Chapter 1: Getting started with automapper

## Remarks

This section provides an overview of what automapper is, and why a developer might want to use it.

It should also mention any large subjects within automapper, and link out to the related topics. Since the Documentation for automapper is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Automapper can be installed from nuget running the following command in Package Manager Console:

```
Install-Package AutoMapper
```

Then it can be included in different files within the project it has been installed to by `using` directives:

```
using AutoMapper;
```

Read Getting started with automapper online: https://riptutorial.com/automapper/topic/5254/getting-started-with-automapper

# Chapter 2: Profiles

## 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)`

## Parameters

| Parameter | Details |
|---|---|
| assemblyNamesToScan | Assembly names containing profiles to load and scan. |
| assembliesToScan | Assemblies containing profiles to scan. |
| typesFromAssembliesContainingProfiles | Types from assemblies containing profiles to load and scan |

## Examples

### Basic Profile

Profiles permit the programmer to organize maps into classes, enhancing code readability and maintainability. Any number of profiles can be created, and added to one or more configurations as needed. Profiles can be used with both the static and instance-based APIs.

```
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);
    }
}
```

## Loading all profiles in an assembly

Often it is useful to be able to load all of the profiles in one or more assemblies into a configuration. AutoMapper provides the method `AddProfiles` method, which has several overloads that allow profiles to be loaded by passing the Assembly, specifying the assembly name, or specifying a type contained in the assembly. Only classes inheriting from **AutoMapper.Profile** will be located and added to the configuration.

Load all profiles in an assembly by specifying the name of the assembly:

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

Load all profiles in an assembly by specifying a type from the assembly:

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

Load all profiles in the current assembly:

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

Read Profiles online: https://riptutorial.com/automapper/topic/5729/profiles

# Credits

| S. No | Chapters | Contributors |
|---|---|---|
| 1 | Getting started with automapper | Community, meJustAndrew |
| 2 | Profiles | Brett Wolfington, Tom |