ASP.NET Core 2.2: Unable to resolve service for type 'AutoMapper.IMapper'

1

★

2

ASP.NET Core (Version: 2.2.102)

I am building an API to return Portos and Especies, but anytime that I access /api/portos (as defined in the controller), I get this error:

> InvalidOperationException: Unable to resolve service for type 'AutoMapper.IMapper' while attempting to activate 'fish.Controllers.PortosController'.
>
> Microsoft.Extensions.DependencyInjection.ActivatorUtilities.GetService(IServiceProvider sp, Type type, Type requiredBy, bool isDefaultParameterRequired)

I am not sure what am I doing wrong, so any help is appreciated.

Models

Especie.cs

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace fish.Models
{
    [Table("Especies")]
    public class Especie
    {
        public int Id { get; set; }
        [Required]
        [StringLength(255)]
        public string Nome { get; set; }

        public Porto Porto { get; set; }
```

```csharp
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel.DataAnnotations;

namespace fish.Models
{
    public class Porto
    {
        public int Id { get; set; }
        [Required]
        [StringLength(255)]
        public string Nome { get; set; }
        public ICollection<Especie> Models { get; set; }

        public Porto()
        {
            Models = new Collection<Especie>();
        }
    }
}
```

Controller

PortoController.cs

```csharp
using System.Collections.Generic;
using System.Threading.Tasks;
using AutoMapper;
using AutoMapper.QueryableExtensions;
using fish.Controllers.Resources;
using fish.Models;
using fish.Persistence;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace fish.Controllers
{
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

```csharp
            this.mapper = mapper;
            this.context = context;
        }


        [HttpGet("/api/portos")]
        public async Task<IEnumerable<PortoResource>> GetPortos()
        {
            var portos = await context.Portos.Include(m => m.Models).ToListAsync();

            return mapper.Map<List<Porto>, List<PortoResource>>(portos);
        }

    }
}
```

Controller>Resources

PortoResources.cs

```csharp
using System.Collections.Generic;
using System.Collections.ObjectModel;

namespace fish.Controllers.Resources
{
    public class PortoResource
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public ICollection<EspecieResource> Models { get; set; }

        public PortoResource()
        {
            Models = new Collection<EspecieResource>();
        }
    }
}
```

EspecieResource.cs

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

```
        public string Nome { get; set; }
    }
}
```

More Relevant Code

Stratup.cs

```
public void ConfigureServices(IServiceCollection services)
{
        services.AddAutoMapper();

        services.AddDbContext<FishDbContext>(options =>
 options.UseSqlServer(Configuration.GetConnectionString("Default")));

        services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_2);

        // In production, the Angular files will be served from this directory
        services.AddSpaStaticFiles(configuration =>
        {
            configuration.RootPath = "ClientApp/dist";
        });
}
```

MappingProfile.cs

```
using AutoMapper;
using fish.Controllers.Resources;
using fish.Models;

namespace fish.Mapping
{
    public class MappingProfile : Profile
    {
        public MappingProfile()
        {
```

FishDbContext.cs

```
using fish.Models;
using Microsoft.EntityFrameworkCore;

namespace fish.Persistence
{
    public class FishDbContext : DbContext
    {
        public FishDbContext(DbContextOptions<FishDbContext> options) : base(options)
        {

        }

        public DbSet<Porto> Portos { get; set; }
    }
}
```

`c#` `api` `asp.net-core` `.net-core` `asp.net-core-mvc`

asked Jan 17 at 15:56

Goncalo Peres
1,630 1 8 22

A complete answer with an example click this link – Iman Bahrampour May 25 at 6:11

1 Answer

You will have to use automapper package as shown below:

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddAutoMapper();
}
```

Refer [this blog for more details.](#)

EDIT:

There is very nice description [from this thread.](#)

You need to add code like below in startup.cs.

You have missed to add IMapper in DI. Please refer add Singleton call from below code.

```
public void ConfigureServices(IServiceCollection services) {
    // .... Ignore code before this

    // Auto Mapper Configurations
    var mappingConfig = new MapperConfiguration(mc =>
    {
        mc.AddProfile(new MappingProfile());
    });

    IMapper mapper = mappingConfig.CreateMapper();
    services.AddSingleton(mapper);

    services.AddMvc();

}
```

edited Jan 17 at 16:31   answered Jan 17 at 16:00

                                                        Manoj Choudhari
                                                        2,735   2   7   22

---

2   Thank you for your time. I have added that package. Check Startup.cs –   Goncalo Peres   Jan 17 at 16:02 ✎

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook ✕

1   Been a minute since I started a new project and needed to include AutoMapper. Got a depreciation warning when I added services.AddAutoMapper();
      to startup.cs. Your edited answer was extremely helpful. Thank you! – ThaKidd KG5ORD May 5 at 7:40 ✎

Got a question that you can't ask on public Stack Overflow? Learn more about sharing private information with Stack Overflow for Teams.   ✕

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook
                                                                     ✕