

Exemplos de Possibilidades com o MMapper

Exemplos

1. Mapeamento Simples de Objetos

```
public class UsuarioDto  
  
{  
  
    public string Nome { get; set; }  
  
    public int Idade { get; set; }  
  
}
```

```
public class UsuarioModel  
  
{  
  
    public string Nome { get; set; }  
  
    public int Idade { get; set; }  
  
}
```

```
var source = new UsuarioDto { Nome = "Carlos", Idade = 30 };  
  
var destino = mapper.Map<UsuarioDto, UsuarioModel>(source);
```

2. Mapeamento com Renomeação de Propriedade

```
public class ProdutoDto  
  
{  
  
    public string Titulo { get; set; }  
  
}
```

```
public class ProdutoEntity

{

    public string NomeProduto { get; set; }

}


config.CreateMap<ProdutoDto, ProdutoEntity>(cfg =>

{

    cfg.ForMember("Titulo", "NomeProduto");

});


var produto = new ProdutoDto { Titulo = "Notebook Gamer" };

var entidade = mapper.Map<ProdutoDto, ProdutoEntity>(produto);
```

3. Conversão de Tipos - String para Inteiro

```
public class PedidoDto

{

    public string NumeroPedido { get; set; }

}


public class PedidoEntity

{

    public int NumeroPedido { get; set; }

}


var pedido = new PedidoDto { NumeroPedido = "12345" };

var entidade = mapper.Map<PedidoDto, PedidoEntity>(pedido);
```

4. Conversão Automática de byte[] para string

```
public class BlobSource
```

```
{
```

```
    public byte[] Dados { get; set; }
```

```
}
```

```
public class TextoDestino
```

```
{
```

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

```
}
```

```
var blob = new BlobSource { Dados = Encoding.UTF8.GetBytes("Documento secreto") };
```

```
var texto = mapper.Map<BlobSource, TextoDestino>(blob);
```

```
// texto.Dados => "Documento secreto"
```

5. Mapeamento de Objetos Aninhados (Nested Objects)

```
public class ClienteDto
```

```
{
```

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

```
    public EnderecoDto Endereco { get; set; }
```

```
}
```

```
public class EnderecoDto
```

```
{
```

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

```
public class ClienteModel  
  
{  
  
    public string Nome { get; set; }  
  
    public EnderecoModel Endereco { get; set; }  
  
}
```

```
public class EnderecoModel  
  
{  
  
    public string Cidade { get; set; }  
  
}
```

```
config.CreateMap<ClienteDto, ClienteModel>();  
  
config.CreateMap<EnderecoDto, EnderecoModel>();
```

```
var cliente = new ClienteDto { Nome = "Julia", Endereco = new EnderecoDto { Cidade = "São Paulo" } };  
  
var model = mapper.Map<ClienteDto, ClienteModel>(cliente);
```

6. Mapeamento Reverso

```
var destino = mapper.Map<Source, Destination>(source);  
  
var novoSource = mapper.Map<Destination, Source>(destino);
```

7. Conversão de DateTime para string

```
public class Evento
```

```
{  
  
    public DateTime DataEvento { get; set; }  
  
}
```

```
public class EventoDto
```

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

```
var evento = new Evento { DataEvento = DateTime.Now };
```

```
var eventoDto = mapper.Map<Evento, EventoDto>(evento);
```

```
// eventoDto.DataEvento => "2025-04-28T13:45:00" (por exemplo)
```

8. Ignorar Propriedades

Atualmente o IMapper ignora automaticamente propriedades que não possuem correspondência de nomes.

(Sugestão futura: adicionar suporte explícito via Ignore() em ForMember.)

9. Conversões Personalizadas Complexas

```
config.CreateMap<Source, Destination>(cfg =>
```

```
{  
  
    cfg.ForMember("Cpf", "CpfComMascara");
```

```
});
```

```
...
```

Exemplo fictício: aqui você poderia depois aplicar uma lógica extra para mascarar o CPF no mapeamento.

10. Mapeamento de Listas de Objetos

```
List<Source> listaSource = new List<Source>
```

```
{
```

```
    new Source { Nome = "João" },
```

```
    new Source { Nome = "Maria" }
```

```
};
```

```
List<Destination> listaDestino = listaSource
```

```
    .Select(src => mapper.Map<Source, Destination>(src))
```

```
    .ToList();
```