Right now, our web API exposes the database entities to the client. The client receives data that maps directly to your database tables. However, that's not always a good idea. Sometimes you want to change the shape of the data that you send to client. For example, you might want to:

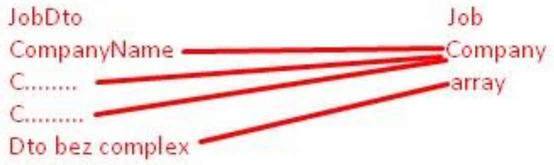
- Remove circular references (see previous section).
- Hide particular properties that clients are not supposed to view.
- Omit some properties in order to reduce payload size.
- Flatten object graphs that contain nested objects, to make them more convenient for clients.
- Avoid "over-posting" vulnerabilities. (See Model Validation for a discussion of over-posting.)
- Decouple your service layer from your database layer.

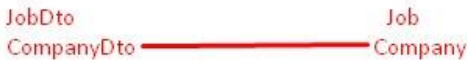
To accomplish this, you can define a *data transfer object* (DTO). A DTO is an object that defines how the data will be sent over the network.

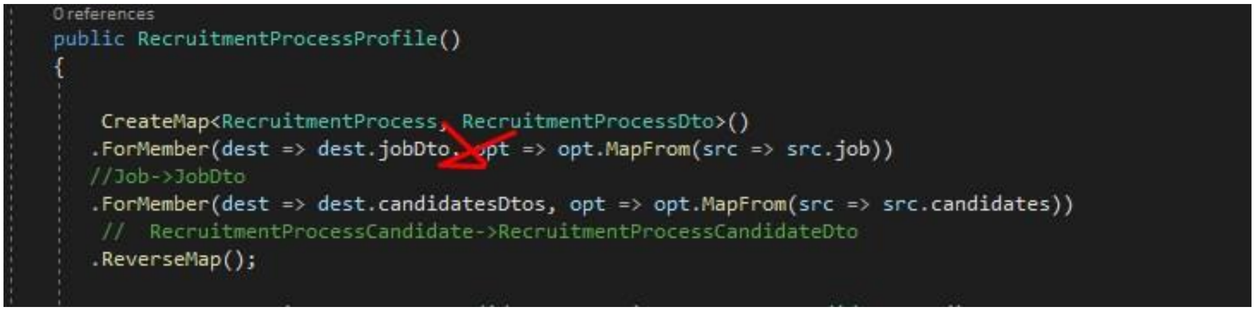
Create Dto with no circular reference, you can create more Dto for one Entity

you can use AutoMapper to fill Dto with data

example mappings but keep an eye for circular dependency(slower api, improper application design)



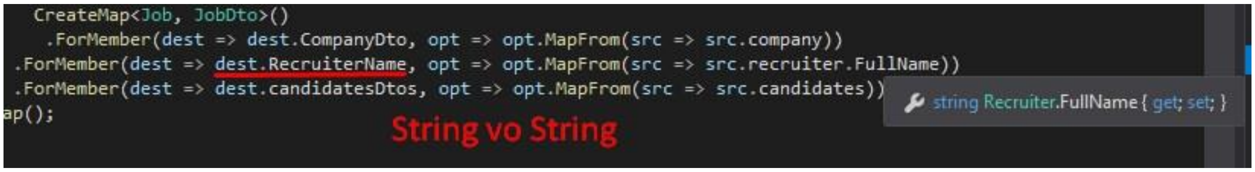




```
0 references
public RecruitmentProcessProfile()
{

    CreateMap<RecruitmentProcess, RecruitmentProcessDto>()
   .ForMember(dest => dest.jobDto, opt => opt.MapFrom(src => src.job))
   //Job->JobDto
   .ForMember(dest => dest.candidatesDtos, opt => opt.MapFrom(src => src.candidates))
    //  RecruitmentProcessCandidate->RecruitmentProcessCandidateDto
   .ReverseMap();
```

se mapira od RecruitmentProcess koj sodrzi job vo jobDto koj se naoga vo RecruitmentProcessDto



```
  CreateMap<Job, JobDto>()
    .ForMember(dest => dest.CompanyDto, opt => opt.MapFrom(src => src.company))
.ForMember(dest => dest.RecruiterName, opt => opt.MapFrom(src => src.recruiter.FullName))
.ForMember(dest => dest.candidatesDtos, opt => opt.MapFrom(src => src.candidates))
ap();
```

string Recruiter.FullName { get; set; }

String vo String

Adding new Entity with Dto if the new entity has relative Data

```
            //zosto mapiram samo fk bidejki drugite
            //ke pravat cicrcular dependence i posle na add ili attach ke dava iskluok
only one instance with given key
            //When attaching existing entities, ensure that only one entity instance
            //with a given key value is attached.


            //dodeka za nazad nema potreba

            CreateMap<RecruiterDto, Recruiter>()
                .ForMember(dest => dest.CompanyFK, opt => opt.MapFrom(src =>
src.companyDto.Id))
             .ForMember(dest => dest.jobs, opt => opt.MapFrom(src => src.jobsDto));



            CreateMap<Recruiter, RecruiterDto>()
                .ForMember(dest => dest.companyDto, opt => opt.MapFrom(src =>
src.company))
                .ForMember(dest => dest.jobsDto, opt => opt.MapFrom(src => src.jobs));
```