1. As resizing array is not possible, it is better to store our data in Lists. Lists offer us more functionality and more flexibility. (with that said I think it is better to assign “roles” and “teams” to a List of strings)
2. All\_lines\_in\_file is not a good naming convention. In c# for local variables and method arguments has to be used “camelCase”. So instead of all\_lines\_in\_file, we must name it allLinesInFiles.
3. If we know the type of the variable, assign the variable to its type, instead of using var. It is better for reading and maintain the code.
4. We can use object initializers for creating objects

Before:

JsonEmployee e = new JsonEmployee();

e.Id = i;

e.Name = all\_lines\_in\_file[i].Split('\t')[0];

e.SurName = all\_lines\_in\_file[i].Split('\t')[1];

e.Email = all\_lines\_in\_file[i].Split('\t')[2];

e.Age = generator.Next(18, 66);

After :

JsonEmployee e = new JsonEmployee

{

Id = i,

Name = all\_lines\_in\_file[i].Split('\t')[0],

SurName = all\_lines\_in\_file[i].Split('\t')[1],

Email = all\_lines\_in\_file[i].Split('\t')[2],

Age = generator.Next(18, 66)

};

1. When you create a string that will be formatted later(you can use interpolation too), it is more readable to start the parameters from 0 and continue to their count. With that said, instead of starting with Id:{7}, it is more readable to start with Id:{0}, ManagerId:{1} and so on…
2. Typo in variable “formattedEmployeed”. Should be formattedEmployees.