

Using ThenInclude with the specification pattern

The Include method we use here can only take a single expression and cannot be used for multiple expressions. One way to achieve this is to use strings to include related entities. So we could add the following to the ISpecification interface:

```
List<string> IncludeStrings {get;}
```

Then the implementation in the BaseSpecification<T> class:

```
public List<string> IncludeStrings { get; } = new List<string>();

protected void AddInclude(string includeString)
{
    IncludeStrings.Add(includeString);
}
```

In the specification evaluator you would add the following:

```
query = specification.IncludeStrings.Aggregate(query,
                                                (current, include) =>
current.Include(include));
```

Then finally in order to use this you would (in your specification) you would pass a string instead of the expression e.g:

```
AddInclude(c => c.CountryOfBirth);
AddInclude("CountyOfOrigin.DefaultCountryLanguage");
```

You can chain on as many ThenIncludes as you wish separating each one with a period.
Obviously you need to be careful with spelling as this is string based.

#TryCatchLearn/Student-faqs