Declare variable within LINQ select(x => new

Ask Question

21

1

I'm mapping a POCO into a model, code shown below.

```
// NOT NEEDED var noneRequiredUserDocuments = new List<NoneRequiredUserDocument>();
//var docs = studentDocuments.Where(x => x.RequiredUserDocumentId == null); // NOT
NEEDED .ToList();
//var noneRequiredUserDocuments = docs.Select(x => new NoneRequiredUserDocument

// You can chain LINQ methods (i.e. Where and Select)
var noneRequiredUserDocuments = studentDocuments
    .Where(x => x.RequiredUserDocumentId == null)
    .Select(x => new NoneRequiredUserDocument
    {
        StudentDocument = x,
        Src = _storageService.GetFileUrl(x.FileName),
        ThumbnailImageUrl = ImageHelper.ThumbnailImageUrl(Src, 75)

    }).ToList();
```

My problem is that in this line:

```
ThumbnailImageUrl = ImageHelper.ThumbnailImageUrl(Src, 75)
```

`Src` doesn't exist in the context.

Is there a way for me to declare a variable within the select that I can the reuse within the LINQ select?

And I don't want to call `_storageService.GetFileUrl` twice.

c# linq

11.9k 14 76 97

asked Mar 25 '15 at 8:50

Michael Esteves
498 3 9 29

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

4 Answers

38

✓

You can declare a variable inside a `Select` like this:-

```
noneRequiredUserDocuments = docs.Select(x =>
  {
      var src= _storageService.GetFileUrl(x.FileName);
      return new NoneRequiredUserDocument
     {
        StudentDocument = x,
        Src = src,
        ThumbnailImageUrl = ImageHelper.ThumbnailImageUrl(
     };
  }).ToList();
```

In `query syntax` doing this is equivalent to:-

```
from x in docs
let src= _storageService.GetFileUrl(x.FileName)
select and so on..
```

edited Mar 25 '15 at 11:32

answered Mar 25 '15 at 8:53

Rahul Singh
19.2k 4 24 42

8

You can use the "let" keyword:

```
var list = (from x in docs
            let temp = _storageService.GetFileUrl(x.FileNa
            select new NoneRequiredUserDocument
            {
                StudentDocument = x,
                Src = temp,
                ThumbnailImageUrl = ImageHelper.ThumbnailI
            }).ToList();
```

answered Mar 25 '15 at 8:55

Tsef

628 3 19

6

you can create a regular code block instead of running a single statement within the lambda expression, this way you can just declare a variable "src" and it will be available throughout the entire block - following regular scope rules.

```
noneRequiredUserDocuments = docs.Select(x => {
      var  src = _storageService.GetFileUrl(x.FileName);
      return
             new NoneRequiredUserDocument
             {
                 StudentDocument = x,
                 Src = src,
                 ThumbnailImageUrl = ImageHelper.Thumbnail
              };
    }).ToList();
```

edited Apr 5 '17 at 12:29

answered Mar 25 '15 at 8:54

Veverke

3,087 24 64

2

You can introduce temporary collection of anonymous type, with all necessary data:

```
noneRequiredUserDocuments = docs
   .Select(x => new { Data = x, Src = _storageService.GetF
   .Select(x => new NoneRequiredUserDocument
   {
       StudentDocument = x.Data,
       Src = x.Src
       ThumbnailImageUrl = ImageHelper.ThumbnailImageUrl(x
   }).ToList();
```

answered Mar 25 '15 at 8:52

Konrad Kokosa

14.3k 2 26 49

Why does he need to use anonymous type if he can use local variable in lambda? – Alex Sikilinda Mar 25 '15 at 8:57

@AlexSikilinda, because it is one of the possibilities to do that – Konrad Kokosa Mar 25 '15 at 8:59

1 Since you can use both the anonymous and strongly typed approach - I think the latter should be the preferred approach. – Veverke Mar 25 '15 at 9:34