## LINQ - Add property to results

Ask Question



Is there a way to add a property to the objects of a Linq query result other than the following?

12







I want to do this without listing out all of the current properties of my object. There are many properties, and I don't want to have to update this code whenever I may change my class.

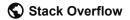
I am still learning with LINQ and I appreciate your suggestions.



asked Feb 25 '09 at 21:02



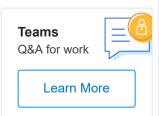
4 Answers



Tags

Users

Jobs



```
public partial class Courses
{
    public String NewProperty { get; set; }
```



Then you can assign it after you've created the object.

answered Feb 26 '09 at 10:44



weiran 699 1 7 17

That's compile time. Isn't the question asking for run time, thus making this not a valid answer to the question? – peSHIr Feb 26 '09 at 11:11

4 Of course, if you're going to alter the original class design (to make it a partial), you might as well just go ahead and add the property... – Eric King Feb 26 '09 at 13:19

The class is generated by LINQ to SQL. This is the way to do it. I don't know why I didn't think to do this! I have done this before. I think it's called a brain fart. — Ronnie Overby Feb 26 '09 at 13:52



I suppose you could return a new object composed of the new property and the selected object, like:

6



answered Feb 25 '09 at 21:08

I was really looking more for a way to "inject" the property into my current object. But thanks! – Ronnie Overby Feb 25 '09 at 21:10



eking's answer will be the most straightforward approach.





If that doesn't work for you (because you need to pass the results around or whatever), and assuming the class you're dealing with already defines the property you want to set, you could create a copy constructor or factory method that takes an existing instance plus the value of the property you want to set:

Alternatively, if Course doesn't define the property, you could subclass it and use the same approach:

(obviously, pick a better name for your subclass)

Again, though, unless you really need to do this, stick with eking's solution.

answered Feb 25 '09 at 21:16





- ServiceStack has a <u>built-in way</u> to handle this with the PopulateWith method.
  - Here's a code example.

```
foreach (var item in results)
{
    var test1 = new ItemDto().PopulateWith(item);
    test1.extraField1 = "extra";
    response.Add(test1);
}`
```

 And if you're not using ServiceStack, you can always use <u>AutoMapper</u>.

```
CreateMap<Foo, Bar>().ForMember(x => x.ExtraBarProperty, opt => opt.Ignore());

edited May 23 '17 at 12:33

Community ◆
1 1

answered Oct 6 '15 at 15:51

Jim G.
9,600 16 83 139
```