Loop Through LINQ Query Columns (Not rows)

Ask Question

5

Is it possible, and if so how, to loop though the results of a LINQ query?

Something like this:

```
var results= from a in dt.AsEnumerable()
                    where a.Field<int>("id") == i
                    select new
                    {
                       id= a.Field<int>("id"),
                       a= a.Field<double>("a"),
                       b = a.Field<double>("b")
                    };

IEnumerable<string> colNames = results.First().GetType().GetProperties()
                                                        .Select(p => p.Name);

string[] columns = colNames.ToArray();

int i = 0;
foreach (var row in results)
{
    for (int i = 0; i < columns.Count(); i++)
    {
        string foobar = (string)row[columns[i]];
        i++;
```

```
    }
}
```

Essentially, i want to replicate the following sort of functionality:

```
DataRow dr = new DataRow();
string foobar = dr[columns[i];
```

Thanks all in advance.

c# linq

edited Nov 22 '14 at 14:22

Yuval Itzchakov
117k 26 178 247

asked Oct 27 '11 at 13:44

CatchingMonkey
1,065 2 12 33

1 You have to cast, so `string foobar = (string)dr[columns[i]; – xanatos Oct 27 '11 at 13:49

maybe this answer will help you stackoverflow.com/questions/1882935/… – Ivan Crojach Karačić Oct 27 '11 at 13:50

@xantos, good point, well made. – CatchingMonkey Oct 27 '11 at 13:53

3 Answers

4

If you have the option of changing your original LINQ statement to produce a data structure that allows you to do what you're looking for, I'd definitely suggest that.

If not, you'll need to use reflection to look up the properties of your anonymous type by their name, and then get the values of those properties by reflection:

```
PropertyInfo[] columns = results.First().GetType().GetProperties
...
        string foobar = columns[i].GetValue(row, null);
```

edited Oct 27 '11 at 13:56

answered Oct 27 '11 at 13:50

StriplingWarrior
110k 20 185 235

I havent really got the option no. However, i have used a bit of reflection to get the properties, i just dont know what to do with them now! – CatchingMonkey Oct 27 '11 at 13:51

@CatchingMonkey: See my update – StriplingWarrior Oct 27 '11 at 13:56

0

Well my answer actually is based on someone's else answer (I will be leaving a link below) with sligh changes: stovroz

So here is the code:

```
foreach(var item in db.Products) {
   System.Reflection.PropertyInfo[] fields = Product.GetType().GetPr
```

```
    foreach(var f in fields) {
        Console.WriteLine(f.Name + ": " + f.GetValue(item));
    }
}
```

Actually I was looking for such a solution to reach dynamic LINQ usage and this guy saved my day. Huge thanks to him!

answered Nov 29 '18 at 9:39

NekoMisaki
26 5

3

```
Action<string> processColumName = (cname) =>
                                {
                                    // do anything you want with a column
                                    string foo = (string) Row[cname];
                                };

results.First()
        .GetType()
        .GetProperties()
        .Select(p => p.Name)
        .ToList().ForEach(processColumName);
```

answered Oct 27 '11 at 13:56

sll
49.6k 14 83 133
