Method to Find GridView Column Index by Name

Ask Question

20

7

I'm trying to write a small method to loop through and find a `GridView` Column by its Index, since it can change position based on what might be visible.

Here is what I have so far:

```
private int GetColumnIndexByName(GridView grid, string name)
{
    foreach (DataColumn col in grid.Columns)
    {
        if (col.ColumnName.ToLower().Trim() == name.ToLower().Trim()) return
col.Ordinal;
    }

    return -1;
}
```

In this case, DataColumn doesn't appear to be the right type to use, but I'm kind of lost as to what I should be doing here.

I can only use .NET 2.0 / 3.5. I can't use 4.0.

c# asp.net

edited Mar 29 '16 at 11:23

Maroun
74.8k 19 146 207

asked Oct 13 '10 at 15:13

Delebrin
764 3 9 19

7 Answers

0

Here's a VB version

```
Protected Function GetColumnIndexByHeaderText(grid As GridView, findH
Integer
    Dim i As Integer = 0
    For i = 0 To grid.Columns.Count - 1
        If grid.Columns(i).HeaderText.ToLower().Trim() = findHeader.
Then
            Return i
        End If
    Next

    Return -1
End Function
```

answered Dec 16 '18 at 16:27

Eric Barr
2,508 1 21 36

0

In case if you need a column itself and not just its index you can use some Linq magic:

```
DataControlField col=GridView1.Columns.Cast<DataControlField>().Firs
== "Column_header")
```

answered Mar 16 '18 at 11:44

montonero

632 3 9

0

```
//Get index of column by header text.
    int GetColumnIndexByName(GridViewRow row, string headerText)
    {
        int columnIndex = 0;
        foreach (DataControlFieldCell cell in row.Cells)
        {
            if(cell.ContainingField is TemplateField){
                if(((TemplateField)cell.ContainingField).HeaderText.
                {
                    break;
                }
            }
          if(cell.ContainingField is BoundField){
                    if
(((BoundField)cell.ContainingField).HeaderText.Equals(headerText))
                {
                    break;
                }
          }
            columnIndex++;
        }


        return columnIndex;
    }
```

answered Jun 19 '17 at 11:21

shefter

1

Home

PUBLIC

I figured it out, I needed to be using `DataControlField` and slightly different syntax.

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work

Learn More

33

The working version:

```
private int GetColumnIndexByName(GridView grid, string name)
    {
        foreach (DataControlField col in grid.Columns)
        {
            if (col.HeaderText.ToLower().Trim() == name.ToLower().Tr
            {
                return grid.Columns.IndexOf(col);
            }
        }

        return -1;
    }
```

edited Mar 29 '16 at 11:23
Maroun
74.8k 19 146 207

answered Oct 13 '10 at 15:30
Delebrin
764 3 9 19

This way, works for me (.NET Gridview):

0

```
    private int GetColumnIndexByName(GridView grid, string name)
    {
        for (int i = 0; i < grid.HeaderRow.Cells.Count; i++)
        {
            if (grid.HeaderRow.Cells[i].Text.ToLower().Trim() == nam
            {
                return i;
            }
        }
        return -1;
    }
```

answered Oct 9 '14 at 12:21

Marcos Paulo Soares
9 1

It would appear that `HeaderRow` isn't initialised until after data is bound. – Matt Jun 12 '16 at 23:41

Correction: until after `DataSource` is set. – Matt Jun 13 '16 at 0:09

2

Better solution which works for Datafield, SortExpression and headerText.

```
public static int GetBoundFieldIndexByName(this GridView gv,string na
    {
        int index = 0;
        bool found = false;
        foreach (DataControlField c in gv.Columns)
        {
            if (c is BoundField)
            {
                BoundField field = (BoundField)c;
                if (name == field.DataField ||
                    name == field.SortExpression ||
                    name == field.HeaderText)
                {
                    found = true;
                    break;
                }
            }
            index++;
        }
        return found ? index : -1;
    }
```

answered Oct 7 '14 at 13:04

Hiren Shah
21 1

17

I prefer collection iteration but why bother with the overhead of `foreach` and `grid.Columns.IndexOf` call in this case? Just iterate through array with an index.

```
private int GetColumnIndexByName(GridView grid, string name)
{
    for(int i = 0; i < grid.Columns.Count; i++)
    {
        if (grid.Columns[i].HeaderText.ToLower().Trim() == name.ToLo
        {
            return i;
        }
    }

    return -1;
}
```

edited Dec 12 '12 at 10:47
Răzvan Flavius Panda
16.4k 12 87 139

answered Aug 27 '12 at 18:29
David Sopko
2,297 22 27