

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

Passing multiple argument through CommandArgument of Button in Asp.net

[Ask Question](#)

▲
23
▼ I have a gridview with multiple rows, each has a Update button and I need to pass 2 values when someone clicks on Update button. Aside from packing the arguments inside CommandArgument separated by commas (archaic and not elegant), how would I pass more than one argument?

★
11

```
<asp:LinkButton ID="UpdateButton" runat="server" CommandName="UpdateRow"
CommandArgument="arg_value" Text="Update and Insert" OnCommand="CommandButton_Click" >
</asp:LinkButton>
```

As a note, the values can't be retrieved from any controls on the page, so don't offer any design solutions. Please concentrate on the question asked. Thanks in advance.

asp.net gridview commandargument

asked Mar 5 '10 at 19:11



sarsnake

10.2k 54 153 272

7 The 'archaic' solution of using a delimiter is sometimes the easiest. – Kelsey Mar 5 '10 at 19:19

@sarsnake - i would accept Archana Motagi's answer – fubo Aug 24 '15 at 12:44

7 Answers



After poking around it looks like Kelsey is correct.

14

Just use a comma or something and split it when you want to consume it.



edited Nov 27 '13 at 7:58



Sid M

3,744 4 22 44

answered Mar 5 '10 at 19:56



cazlab

677 5 17

4 found this: aspdotnetcodebook.blogspot.com/2008/09/... – sarsnake
Mar 5 '10 at 20:29



If you want to pass **two values**, you can use this approach

0

```
<asp:LinkButton ID="RemoveFroRole" Text="Remove From Role" runat="se  
CommandName='<%# Eval("UserName") %>' CommandArgument='<%# Eval("RoL  
OnClick="RemoveFromRole_Click" />
```



Basically I am treating {CommmandName,CommandArgument} as key value. Set both from database field. You will have to use OnClick event and use OnCommand event in this case, which I think is more clean code.

answered Sep 19 '16 at 8:43



Hammad Khan

9,361 12 87 112

Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work



Learn More



13



@Patrick's answer is a good idea, and deserves more credit!. You can have as many data items as you want, they are all separated, and can be used client side if necessary.

They can also be added declaratively rather than in code. I just did this for a GridView like this:

```
<asp:TemplateField HeaderText="Remind">
  <ItemTemplate>
    <asp:ImageButton ID="btnEmail"
      data-rider-name="<%= ((Result)((GridViewRow) Container).Data.
      data-rider-email="<%= ((Result)((GridViewRow) Container).Data.
    %>"
      CommandName="Email" runat="server" ImageAlign="AbsMiddle"
      ImageUrl="~/images/email.gif" />
  </ItemTemplate>
</asp:TemplateField>
```

In the RowCommand, you do this:

```
void gvMyView_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "Email")
    {
        var btnSender = (ImageButton)e.CommandSource;
        var riderName = btnSender.Attributes["data-rider-name"];
        var riderEmail = btnSender.Attributes["data-rider-email"];
        // Do something here
    }
}
```

So much cleaner than hacking all the values together with delimiters and unpacking again at the end.

Don't forget to test/clean any data you get back from the page, in case it's been tampered with!

answered Jul 24 '15 at 11:03



cyberspy

707 8 12

-
- 1 I like this approach, it's as close as you'll get to having the items strongly typed, using commas and substring to separate this kind of info, although workable, can become a mess. You need have inherent knowledge of what each part of the argument is, using "data-" containers is safer and makes the code much more readable. – [GJKH](#) Feb 27 '16 at 16:33
-



5

My approach is using the attributes collection to add HTML data-attributes from code behind. This is more inline with jquery and client side scripting.



```
// This would likely be done with findControl in your grid OnItemCreated  
LinkButton targetBtn = new LinkButton();
```

```
// Add attributes  
targetBtn.Attributes.Add("data-{your data name here}", value.ToString());  
targetBtn.Attributes.Add("data-{your data name 2 here}", value2.ToString());
```

Then retrieve the values through the attribute collection

```
string val = targetBtn.Attributes["data-{your data name here}"].ToString();
```

answered Mar 25 '13 at 15:16



Patrick Boudreau

51 1 1

A little more elegant way of doing the same adding on to the above

3

comment ..

```

<asp:GridView ID="grdParent" runat="server" BackColor="White" Border
    AutoGenerateColumns="false"
    OnRowDeleting="deleteRow"
    GridLines="Vertical">
    <asp:BoundField DataField="IdTemplate" HeaderText="IdTemplate"
    <asp:BoundField DataField="EntityId" HeaderText="EntityId" />
    <asp:TemplateField ShowHeader="false">
        <ItemTemplate>
            <asp:LinkButton ID="lnkCustomize" Text="Delete" CommandName=
CommandArgument='<%=#Eval("IdTemplate") + ";" + Eval("EntityId")%>' r
            </asp:LinkButton>
        </ItemTemplate>
    </asp:TemplateField>
</asp:GridView>

```

And on the server side:

```

protected void deleteRow(object sender, GridViewDeleteEventArgs e)
{
    string IdTemplate= e.Values["IdTemplate"].ToString();
    string EntityId = e.Values["EntityId"].ToString();

    // Do stuff..
}

```

edited Oct 29 '12 at 13:04



Uwe Keim

27.7k 32 135 216

answered Jan 9 '12 at 6:28



Nemo

35 1 4

This only works if you want the IdTemplate and EntityId fields displayed in your grid. The values you places in the CommandArgument is never actually used. – [cyberspy](#) Jul 24 '15 at 11:08

73

You can pass semicolon separated values as command argument and then split the string and use it.

```
<asp:TemplateField ShowHeader="false">
  <ItemTemplate>
    <asp:LinkButton ID="lnkCustomize" Text="Customize" CommandName="Customize" CommandArgument='<#Eval("IdTemplate") + ";" + Eval("EntityId")%>' r
    </asp:LinkButton>
  </ItemTemplate>
</asp:TemplateField>
```

at server side

```
protected void gridView_RowCommand(object sender, GridViewCommandEventArgs e)
{
    string[] arg = new string[2];
    arg = e.CommandArgument.ToString().Split(';');
    Session["IdTemplate"] = arg[0];
    Session["IdEntity"] = arg[1];
    Response.Redirect("Samplepage.aspx");
}
```

Hope it helps!!!!

edited Jul 1 '11 at 23:05



mrk

4,443 3 19 41

answered Jul 1 '11 at 14:44



Archana Motagi

739 1 5 2

The above worked for me. Only problem is you're missing the parameters in the Redirect() ASPX link. Try Response.Redirect("Samplepage.aspx?IdTemplate=" + Session["IdTemplate"].ToString().Trim() + "&IdEntity=" + Session["IdEntity"].ToString().Trim()); – [Fandango68](#) Jun 28 '13 at 5:18

Being a pedant you don't need the .ToString(). CommandArgument is

always a string, so: var argument =
((LinkButton)sender).CommandArgument.Split(';'); – [Norbert Norbertson](#)
Nov 9 '17 at 13:13



1

Either store it in the gridview datakeys collection, or store it in a hidden field inside the same cell, or join the values together. That is the only way. You can't store two values in one link.



answered Mar 5 '10 at 19:17

[Brian Mains](#)

44.2k 32 129 233