How to pass a C# variable as a CommandArgument in ASP.NET

Ask Question

0

I am creating dynamically an array, adding a line for each entry of some data. Each entry must have a button to do some stuff with it, but I am stuck at this point. I have this code to create the *ASP.NET* button:

```
<asp:Button ID="buttonyes" class="buttonyes" CommandArgument='<%= request.ID %>'
            CommandName="AcceptRequest" OnClick='<%= ButtonPressed %>' runat="server"/>
```

But when I run it, I have multiple errors saying that:

> The `%=` and `<` or `>` expressions are not valid.

When I click on the button, I want to call a function written in the **index.aspx.cs** file linked with my **index.aspx**, here is `ButtonPressed` function and with a filter in this function, it choose what action it should do:

```
protected void ButtonPressed(Object sender, CommandEventArgs e)
{
    Button btn = (Button)sender;
    switch (btn.CommandName)
    {
        case "AcceptRequest":
            AcceptRequest(btn.CommandArgument.ToString());
            break;
        case "RefuseRequest":
            RefuseRequest(btn.CommandArgument.ToString());
            break;
    }
}
```

The problem is that I don't succeed in passing those *C#* variables as a `CommandArgument` and `CommandName` .

I already tried those syntaxes, following some previous questions:

- `CommandArgument="<%: request.ID %>"`
- `CommandArgument='<%: request.ID %>'`
- `CommandArgument="<%= request.ID %>"`
- `CommandArgument='<%= request.ID %>'`

But none of them worked.

EDIT 1:

As @grek40 said below, the `OnClick` value should be replace like the following:

```
<asp:Button ID="buttonyes" class="buttonyes" CommandArgument='<%= request.ID %>'
            CommandName="AcceptRequest" OnCommand="ButtonPressed" runat="server"/>
```

EDIT 2:

I would like to know how to pass a C# ***local*** variable since those button should be added in a `foreach` loop like so:

```
foreach (Request request in requests)
     {
     %>
         <tr>
              <td><%: request.applicant.Name + " (" + request.applicant.ID + ")" %></td>
              <td><%: request.software.Name %></td>
              <td><%: request.date %></td>
              <td class="actions">
                   <asp:Button ID="buttonyes" class="buttonyes"
```

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

```
CommandName="AcceptRequest" CommandArgument="<# request.Id %>"
OnCommand="ButtonPressed" runat="server"/>
                   <asp:Button ID="buttonno" class="buttonno"
CommandName="RefuseRequest" CommandArgument="<# request.Id %>" OnCommand="ButtonPressed"
runat="server"/>
              </td>
          </tr>
      <%
      }
```

By the way, the code above doesn't work, since, as @Nino said, this only works for global variables.

c# asp.net

edited Aug 18 '17 at 6:17
user5377037
7,900 12 28 61

asked Aug 7 '17 at 11:35
Louis 'LYRO' Dupont
627 1 5 27

3 Instead of `OnClick='<%= ButtonPressed %>'` it should be `OnClick="ButtonPressed"` as far as I know – grek40 Aug 7 '17 at 11:41

Try this `CommandArgument='<%#Eval("ID")%>'` . It will work – Joy Fernandes Aug 7 '17 at 11:41

Can you quote exactly what the error message says? As grek40 says it looks like the `OnClick` is the invalid attribute, not the `CommandArgument` ... – Chris Aug 7 '17 at 11:56

3 Answers

Why not save yourself a whole lot of trouble and switch to a Strongly

3

+50

Typed Repeater Control. You can make it type safe with the `ItemType` property. The value must the the complete namespace to the class `request`

Then you will still be able to access the properties like you have now. But instead of using `request` , you use the fixed name of `Item` .

```
<table border="1">
    <asp:Repeater ID="Repeater1" runat="server" ItemType="Namespace1
        <ItemTemplate>
            <tr>
                <td><%# Item.applicant.Name + " (" + Item.applicant.
                <td><%# Item.software.Name %></td>
                <td><%# Item.date.ToLongDateString() %></td>
                <td class="actions">
                    <asp:Button ID="buttonyes" class="buttonyes"
CommandName="AcceptRequest" CommandArgument='<%# Item.Id %>' OnComma
runat="server" />
                    <asp:Button ID="buttonno" class="buttonno"
CommandName="RefuseRequest" CommandArgument='<%# Item.Id %>' OnComma
runat="server" />
                </td>
            </tr>
        </ItemTemplate>
    </asp:Repeater>
</table>
```

Now you can use the normal way of databinding: `CommandArgument='<%# Item.Id %>'`

Below the full code for testing...

```
public List<request> requests = new List<request>();

protected void Page_Load(object sender, EventArgs e)
{
    //create some dummy data
    for (int i = 0; i < 10; i++)
    {
        request r = new request() { Id = i, date = DateTime.Now };
        r.applicant = new applicant() { Name = "Applicant " + i, ID
        r.software = new software() { Name = "Software " + i };
        requests.Add(r);
        }

        if (!Page.IsPostBack)
        {
            Repeater1.DataSource = requests;
            Repeater1.DataBind();
        }
    }

    public class request
    {
        public int Id { get; set; }
        public DateTime date { get; set; }
        public applicant applicant { get; set; }
        public software software { get; set; }
    }

    public class applicant
    {
        public int ID { get; set; }
        public string Name { get; set; }
    }

    public class software
    {
        public int ID { get; set; }
        public string Name { get; set; }
    }
```

answered Aug 17 '17 at 20:14

VDWWD
25.2k 12 39 56

1

I think you have the command action incorrect.

Most people might think OnClick(), but your method signature is associated with OnCommand()

Change from

```
OnClick='<%= ButtonPressed %>'
```

to

```
OnCommand='<%= ButtonPressed %>'
```

The OnClick signature is (object sender, EventArgs e)... OnCommand signature is (object sender, CommandEventArgs e)... and the expected parameter value should be good.

As for the "OnClick" or "OnCommand", that should be fixed based on the function that will handle the call, such as your "ButtonPressed" method. You do NOT need your "%" wrappers as the function is not changing... your COMMAND NAME can change, but not the method that handles the event...

```
OnClick='ButtonPressed'
```

edited Aug 7 '17 at 12:04

answered Aug 7 '17 at 11:49

DRapp
38.2k 10 58 115

Thanks for your answer, but my problem is a syntax problem and not really linked with incorrect inputs, even if I changed it as you recommanded and that's one more thing I will be aware of. – Louis 'LYRO' Dupont Aug 7 '17 at 11:52

As grek40 mentioned in comments on the question you don't want the method name to be wrapped in `<%= %>` . – Chris Aug 7 '17 at 11:53

@Chris, revised answer. dont parameterize your onclick command, that should be fixed. – DRapp Aug 7 '17 at 12:04

2

you can do it like this:

```
CommandArgument='<%# request.ID %>'
```

That way `CommandArgument` will be assigned during DataBidning phase.

answered Aug 7 '17 at 11:40

Nino
5,643 2 18 32

Thanks for your answer, but when I try your code, it raise an error saying that the `request` value doesn't exist. But this line is called in a loop which will display all the buttons I need, and I am 100% sure my `request` variable is declared. – Louis 'LYRO' Dupont Aug 7 '17 at 11:43

@Louis'LYRO'Dupont `request` should be defined on aspx/ascx scope. If `request` is local variable, used in some method, then this won't work. Why don't you then dynamically assign `CommandArgument` in code behind? No need to set it in markup... – Nino Aug 7 '17 at 11:48