

How do you access ViewState collection from PreviousPage on cross-page postback?

Asked 16 years, 2 months ago Modified 13 years, 5 months ago

Viewed 4k times



3

In ASP.net 2.0, the PreviousPage property of a web page does not have a ViewState collection. I want to use this collection to transfer information between pages.



asp.net



Share

Improve this question

Follow

asked Oct 3, 2008 at 20:47



[Keith Walton](#)

5,364 ● 8 ● 41 ● 54

4 Answers

Sorted by:

Highest score (default)



1

View State is exclusive to the page.

If you want to transfer items,



- you can persist the data in a database, file, forms auth ticket or other cookie (Dont use Session or HttpContext.Current.Cache if you can help it)





- do a cross page post - from your first page, post back to the second page (and get the details from `HttpContext.Current.Request.Form[]` collection)
- put the values in a query string

Share Improve this answer

answered Oct 3, 2008 at 20:54

Follow



[StingyJack](#)

19.4k ● 11 ● 67 ● 126

I ended up using a cross page post, but accessing a public property on the first page (see below). – [Keith Walton](#) Oct 3, 2008 at 22:29



Use `HttpContext.Current.Items` instead...`ViewState` is only good for the page it is on.

1

Share Improve this answer

answered Oct 3, 2008 at 20:47

Follow



[FlySwat](#)

175k ● 75 ● 248 ● 314



This method is simple and clean. I like it. – [Keith Walton](#)

Oct 3, 2008 at 21:09

I tried to use this method, but I am posting directly from a link on the first page. No code is executing where I can populate the HttpContext.Current.Items collection: <asp:linkbutton runat="server" id="Upload" text="Upload" postbackurl="~/uploadfeescheduleterms.aspx" />

– [Keith Walton](#) Oct 3, 2008 at 22:24



1

You can avoid using PreviousPageType directive, by using some base page class that can hold your object. For example you have class



```
public class BaseCrossPage: System.Web.UI.Page
{
    public List<Guid> Invitees = new List<Guid>();
}
```

So if first page derive from this class

```
public partial class Default : BaseCrossPage
{
    protected void Page_Load(object sender, EventArgs
    {
        this.Invitees = LoadInvitees();
    }
}
```

Then the page that you have posted to can access that object, assuming that previous page derived from BaseCrossPage...

```
public partial class secondPage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs
    {

        BaseCrossPage p = (BaseCrossPage)PreviousPage;
        List<Guid> Invitees = p.InvitedTeams

    }
}
```

kind of "viewstate" between pages...

Share Improve this answer

answered Jul 7, 2011 at 11:06

Follow



jekcom

2,095 ● 2 ● 25 ● 38



0

You can't directly. (See [http://msdn2.microsoft.com/en-us/library/ms178139\(vs.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms178139(vs.80).aspx))

Here's what you can do -



Create public properties on the first page exposing the information you want to share. On the second page, set the PreviousPageType to the first page in the header of aspx file:



```
<%@ previouspagetype virtualpath="~/firstpage.aspx" %>
```

Then, get the values of these properties in the Load event of the second page:

```
If (Not MyBase.IsPostBack) Then  
    _someValue = Me.PreviousPage.SomeValue  
End If
```

Share Improve this answer

edited Dec 2, 2008 at 17:29

Follow

answered Oct 3, 2008 at 21:01



[Keith Walton](#)

5,364 ● 8 ● 41 ● 54

Use the @ PreviousPageType directive to get strong typing against the PreviousPage property. "_someValue = firstPage.SomeValue" becomes "_someValue = Me.PreviousPage.SomeValue" – [Keith Walton](#) Oct 8, 2008 at 16:58
