Re: Determining which Button was pressed

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005−02/6801.html

---

From: Keith Patrick (*richard_keith_patrick_at_hotmail.com*)
Date: 02/22/05

Date: Tue, 22 Feb 2005 16:40:58 −0600

Well, my problems with ASP.Net go much deeper than simple OO concepts (I've got at least 3 different threads addressing 3 different multi−day issues within a single functional problem). I've run into problems with manually calling a request while staying in an existing session. I've run into problems using ASP.Net to simply POST data (and searching on that shows more than a few people have issues with it). I've run into ASP not using its own Request params consistently. I've run into problems with having the System.Net HTTP stuff interact with the System.Web HTTP stuff (the parallel namespace/class stuff drives me NUTS). Maybe it's OO stuff or API design at its heart, but honestly, I haven't run into these issues with the design of the API or its behavioral quirks anywhere to this degree anywhere else in the BCL. And I've also been further frustrated in that in researching solutions, I too often find "Oh, yeah, that's being fixed in ASP.Net 2.0" That'd be all fine and dandy if ASP.Net 2.0 was coming out soon or those fixes would be merged over into a 1.1 service pack, but it's not, and I'm left wondering if I should wait on the fabled fix (and after getting burned holding development on an object layer for the at−one−time impending ObjectSpaces, I have no faith in the "release date fix") or devise some scheme of hacks to get around it or tell the customer, "Uhh, it seems the best solution from newsgroups/blogs is to run ASP.Net 2.0 beta, even though running beta software in production is terrible and all"
I need some magic pill that zonks me out into Betaland.

"Steve C. Orr [MVP, MCSD]" <Steve@Orr.net> wrote in message news:eYUBAfSGFHA.1260@TK2MSFTNGP12.phx.gbl...
> We haven't been talking about much more than basic object oriented
> concepts here, so you're likely to encounter similar issues in virtually
> any modern language or platform.
> However, the ASP.NET page lifecycle does tend to throw some complex timing
> issues into the mix, so perhaps your frustrations are centered around
> that.
> Here are some links that may help in that regard:
>
http://msdn.microsoft.com/library/default.asp?url=/library/en−us/cpguide/html/cpconcontrolexecutionlifecycle.asp
> http://www.15seconds.com/issue/020102.htm
>
http://msdn.microsoft.com/library/default.asp?url=/library/en−us/vbcon/html/vbconWebFormsPageProcessingStages.

>
> --
> I hope this helps,
> Steve C. Orr, MCSD, MVP
> http://SteveOrr.net
>
>
> "Keith Patrick" <richard_keith_patrick@hotmail.com> wrote in message
> news:ON8DSbSGFHA.1296@TK2MSFTNGP10.phx.gbl...
>> Yeah, that looks to be about my only option. It's odd...I've been
>> running into more walls with regards to ASP.Net, whether it be
>> consistency (or lack thereof) or a ridiculous # of hoops to get through
>> to do some simple HTTP things, and it has been absolutely driving me
>> nuts. I guess after being so productive with it for so long, I just
>> expected it to expose certain things to make more advanced things
>> possible (a form of an inductive UI, if you will), but the deeper I dig,
>> the more I get let down with regards to my expectations.
>>
>> "Steve C. Orr [MVP, MCSD]" <Steve@Orr.net> wrote in message
>> news:%235gDhSSGFHA.2756@TK2MSFTNGP15.phx.gbl...
>>> The user control could expose a public property that specifies the
>>> button that was clicked.
>>>
>>> --
>>> I hope this helps,
>>> Steve C. Orr, MCSD, MVP
>>> http://SteveOrr.net
>>>
>>>
>>> "Keith Patrick" <richard_keith_patrick@hotmail.com> wrote in message
>>> news:%23pZ$eOSGFHA.1396@tk2msftngp13.phx.gbl...
>>>>I have a web page that has a bunch of usercontrols, each declaring a
>>>>Button called "GoButton". Now, my parent page has to be able to
>>>>determine which button triggered the event. The catch is: I'd like to
>>>>use the same algorithm for every control, and right now, my algorithm is
>>>>"sender = this.FindControl(Request["__EVENTTARGET"]", but the problem is
>>>>that those buttons always get rendered as type="submit", even if I
>>>>assign a CommandName and/or CommandArgument, leading the
>>>>Request["__EVENTTARGET"] for Button presses to be String.Empty.
>>>>
>>>> The event itself has the sender, but that code is in the usercontrol,
>>>> not the page. What this means for me is that I could hack in a
>>>> solution via temporary variables or passing a sender ID in the
>>>> QueryString, but I'd much rather find a way to crack the Button
>>>> rendering and just have it render as type="button" where
>>>> onclick="javascript:__doPostBack(...)" just like every other control
>>>> I've got to handle.
>>>>
>>>
>>>
>>

>>
>
>