

microsoft.public.dotnet.framework.aspnet: Re: [ENTER] key causing major headaches.

Re: [ENTER] key causing major headaches.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2004-06/6079.html>

From: Eric (*ejproductions-msnews7831_at_mailblocks.com*)

Date: 06/25/04

Date: Fri, 25 Jun 2004 11:40:25 -0700

Well Everyone,

I believe I've stumbled upon an acceptable solution. For the curious-minded...

An obvious solution would be to disable the [ENTER] button entirely. Doing this with javascript however, is a problem because if the client has javascript disabled or doesn't support it you will still get the same "ghost postbacks" where the source (clicked button) is unknown.

My solution is to do this with an image instead of a button. To do this, create a <asp:LinkButton> object and set its Text property to be the HTML for an . You can then use the Click event of the LinkButton object. I didn't think this would work for quite a while (because when I tested it, it didn't work). As it turns out, the reason my tests resulted in failure is because of bug in Internet Explorer...

It turns out that if you have a <form> with less than two text boxes IE automatically submits the form when the [ENTER] key is pressed. This happens WHETHER OR NOT you have a submit button in the <form>. If you have 2 OR MORE text boxes, IE will NOT automatically submit the form when [ENTER] is pressed. Instead, it will CLICK the first button it encounters following the widget with the focus. This latter behavior is what *should* occur regardless of the number of text boxes. So... to use my <asp:LinkButton> solution, you ALSO have to have at least 2 text boxes in the <form>. If your application only calls for one (or zero), you must supplement the "missing" text box(es) with hidden text box(es). This could be accomplished by using CSS to set the visibility to hidden, the width to 0, or both.

Jumping through these hoops will give you the expected behavior. My next step will be to derive my own "ImageButton" control (from asp:LinkButton) that includes the hidden text boxes. It may also accept an alternate MouseoverImage property to emulate the appearance of an actual button (using *optional* client-side javascript).

If some jaded soul out there winds up as stuck as I was, I hope this solution finds them!

Re: [ENTER] key causing major headaches.

microsoft.public.dotnet.framework.aspnet: Re: [ENTER] key causing major headaches.

Regards,

Eric

"Eric" <ejproductions–msnews7831@mailblocks.com> wrote in message
news:exRi9atWEHA.3716@TK2MSFTNGP11.phx.gbl...

> *Hi All,*

>

> *I'm very experienced in traditional ASP and am new to (am learning)
ASP.NET.*

> *FYI: I am initially learning ASP.NET with VB.NET to ease the transition
for*

> *me. I have encountered what I believe to be a huge limitation with
ASP.NET*

> *and I'm wondering how you have handled this problem.*

>

> *Let's suppose you have a page where the user can do one of two things: add
a*

> *"Category", or add a "Movie Title" to an already existing category. This
is*

> *just a make-believe scenario that I'm using to make a point: You have two*

> *distinct sections to your page... one section contains a text box and a*

> *button (for adding a new category) and another section contains a dropdown*

> *menu of categories, a text box, and a different button (for adding a new*

> *movie title).*

>

> *You can have have the Click event of the first button ("bAddCategory")*

> *handle the addition of the new category, and you can have the Click event
of*

> *the second button ("bAddMovie") add the new movie title. So far, so good.*

>

> *My problem is this: what if the user presses the [ENTER] key on his/her*

> *keyboard? Doing this does not fire either Click event. I can (in
essence)*

> *make a "default" button by adding the following to the server-side Load*

> *event for the page:*

>

> *Page.RegisterHiddenField("__EVENTTARGET", bAddMovie.ClientID)*

>

> *Doing this will cause the Click event for bAddMovie to fire when the user*

> *presses [ENTER]. However... I don't want bAddMovie to be the default*

> *button*

> *if the user is in the process of typing a new category name (in that case,*

I

> *obviously would want bAddCategory to be the default).*

>

> *In traditional ASP, I would simply have two <form>s on the page. One*

> *<form>*

> *would encapsulate the widgets for entering a category and a second <form>*

> *would encapsulate the widgets for entering a movie. Each of the two*

> *<form>*

Re: [ENTER] key causing major headaches.

microsoft.public.dotnet.framework.aspnet: Re: [ENTER] key causing major headaches.

> tags would contain a `<input type="hidden">` element containing a name/value
> pair indicating which form was submitted on postback.

>

> In ASP.NET, I see no way to emulate this concept. ASP.NET does not let me
> have two form tags (with `runat="server"` defined). And, if I use a
> client-side (traditional) `<form>`, I cannot use any of the ASP.NET web
> controls (like `<asp:DropDownList>`, etc.). Not being able to use these web
> controls, to me, defeats the whole purpose of using ASP.NET. If I revert
> back to a traditional `<form>` tag, I don't get events, viewstates, or any
> other fancy ASP.NET features.

>

> This concept (of having two forms) is extremely important to me. What do

I

> do!? Any and all help is very much appreciated!!

>

>

> Regards,

>

> Eric

>

>