

Re: Inhibiting Browser Back/Fwd Buttons via PHP?

Source: <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2003-10/1536.html>

From: Ralph Freshour (ralph_at_primemail.com)

Date: 10/20/03

Date: Mon, 20 Oct 2003 13:55:31 GMT

First off, I'd like to say that the browser back and fwd buttons were implemented when web pages were **static** – but as time and technology went on, dynamic web pages came into being and we all know how critical it is to call dynamic web pages via the links on web each page, not by clicking the back/fwd buttons.

Now to respond to your reply: I don't want to display an expired page because thats like displaying an error message – the user may continue to click on the back button and then at the point the pages are all screwed up from a dynamic data stand point.

I don't mind making the buttons disappear and because I tell them in my help page not to use the back/fwd buttons, if they work around that somehow and do invoke the buttons in another manner, I'll tell then not to do that when they email me – how do you remove those buttons or otherwise make them inoperative?

Thanks...

On 20 Oct 2003 06:17:07 –0700, tony@marston-home.demon.co.uk (Tony Marston) wrote:

>*Keith Bowes* <do.not@spam.me> wrote in message news:<1066549704.905234@cache1>...

>> *Ralph Freshour* wrote:

>> > *Is it possible to inhibit the browser Back/Fwd buttons via PHP?*

>> >

>>

>> *No. Firstly, PHP is a server-side technology and could only be used to generate the appropriate client-side code.*

>

>*Yes, it is possible, because I have done it. It involves the use of sessions and self-executing scripts (ones where the script that outputs a page via the GET method also deals with the corresponding POST method). The technique is as follows:–*

>

>*a) Within a form any link to a different form does not invoke that form directly, instead it sends a message to the current form containing the identity of the required form.*

comp.lang.php: Re: Inhibiting Browser Back/Fwd Buttons via PHP?

- >b) The required form is validated as some users may not be able to*
- >access some forms. If it is valid then it stores the new script name*
- >in a session variable called `$_SESSION['expected_script']`, then passes*