

Re: [PHP] Passing variables between pages

Source: <http://coding.derkeiler.com/Archive/PHP/php.general/2008-09/msg01272.html>

- *From:* eric.butera@xxxxxxxx ("Eric Butera")
 - *Date:* Mon, 22 Sep 2008 10:18:42 -0400
-

On Mon, Sep 22, 2008 at 9:49 AM, tedd <tedd.sperling@xxxxxxxx> wrote:

At 11:19 PM +0100 9/21/08, Nathan Rixham wrote:

so may as well just have everything in one script then..

You always have the option to write one huge script for any application.

But normally in programming, we confront any problem by breaking it down into smaller steps and writing code to solve smaller problems. Whenever any step is finished, then we pass the results on to the next step and continue solving the problem -- is this not correct?

In PHP, we use sessions to pass variables between scripts -- but we could just as well use:

```
ob_clean();
include('next.php');
exit(0);
```

For this will continue the program flow with all the variables intact the same as using sessions -- is this not correct?

Maybe I'm not understanding the limitations of this technique -- so, let me put this another way, show me an example where sessions is the answer and my include method won't work.

Cheers,

tedd

--

<http://sperling.com> <http://ancientstones.com> <http://earthstones.com>

--

PHP General Mailing List (<http://www.php.net/>)

Re: [PHP] Passing variables between pages

To unsubscribe, visit: <http://www.php.net/unsub.php>

I don't understand what sort of point you're trying to make with this? The original poster asked how to keep state between different pages which in that context did mean separate requests. It seems like you're trying to be confusing on purpose.

In all of my applications there is a clear definition between request variables, session variables, global configuration variables, and local variables. So just defining \$foo saying include 'steptwo.php' to echo \$foo really wouldn't make a whole lot of sense to me from your example. Where did \$foo come from when steptwo.php runs? Do you always assume steptwo.php is an include and prevent direct access to it? Do you check to see if \$foo exists and if not check the query string? Was \$foo validated to be the correct data type/length requirements, etc if so? That would be a double check for nothing in some cases.

I do use this technique inside of my templating class though. All it does is assign variables to a class and then I access them inside of the template which is just included inside of my render() method. This is a very controlled situation though since I know that the context is output and my input has been validated at this point and all output will be escaped.

.