Subject: Maintaining state
Posted by rafs on Wed, 08 Jul 2015 20:10:35 GMT
View Forum Message <> Reply to Message

It seems that Radicore saves the state of the page before going down into a sub-page, then loads the saved state upon return. I am wondering how common this technique is in other webapp frameworks. I took a webapp development class a decade ago and I seem to recall that we saved state by passing parameters into subpages, but I can't be sure.

Subject: Re: Maintaining state
Posted by rafs on Thu, 09 Jul 2015 03:39:05 GMT
View Forum Message <> Reply to Message

Also, it would be helpful to me to see exactly what is being passed from one page to another. I could use a packet sniffer to check this out, but wonder if you haven't already written-up an explanation somewhere. (I see the UML diagrams you have, but it would be helpful to see the data that gets passed, and even how that relates to "$where" vs. $this->sql_?? statements.)

Subject: Re: Maintaining state
Posted by AJM on Thu, 09 Jul 2015 09:49:21 GMT
View Forum Message <> Reply to Message

I do not pass state in the URL as this is a MASSIVE security problem. A naughty user could change the URL and really screw things up.

I use the $_SESSION superglobal to maintain state between requests. There is a brief description at http://www.tonymarston.net/php-mysql/menuguide/appendixg.htm l

The $where variable is a string which be passed from one script to another.

The $this->sql_* variables are used to modify the SQL query which is executed in the current script.

Subject: Re: Maintaining state
Posted by AJM on Wed, 12 Aug 2015 10:08:14 GMT
View Forum Message <> Reply to Message

There is more information in Working with a stateless protocol which you may find interesting.

Subject: Re: Maintaining state
Posted by rafs on Thu, 27 Aug 2015 01:46:27 GMT
View Forum Message <> Reply to Message

I am back to this. I have read about your use of $_SESSION superglobal to pass 'state'.
What is your advice on the best way to view the state info as each new script (page) is called?

I want to see what the $where contains that is being called for the new page.
The technique I am testing (with Eclipse) is to set a breakpoint just after the standard transactions controllers at the initSession().

There under $_SESSION >> 'pages', I see my new (child) page that is being called with a few items: task_id, task_array, pattern_id, 'mypage(link1)', but I am looking for the element that I selected in the parent, 'gid' = '1', but it is not there.

Subject: Re: Maintaining state
Posted by AJM on Thu, 27 Aug 2015 10:08:36 GMT
View Forum Message <> Reply to Message

If you look at the initSession() function inside include.session.inc you will see that it looks for an entry in $_SESSION['pages'][$PHP_SELF]), and then extracts $_SESSION['pages'][$PHP_SELF][$task_id] into a global variable called $script_vars. The contents of $script_vars is then extracted into individual variables such as $where, $search, et cetera.

There is a separate copy of $script_vars for each script and this is is constructed BEFORE the script is called by the calling script. This means that if script "A" calls script "B" then $script_vars for "B" will be constructed by "A" using the scriptNext() function, and this will include any arguments that need to be passed from "A" to "B". When control is passed from "A" to "B" the $script_vars for "A" will remain in $_SESSION so that when control is returned to "A" it will be able to resume from where it left off. When "B" terminates the scriptPrevious() function will cause $script_vars for "B" to be dropped (unless the "Keep data" option on MNU_TASK is set), update the $script_vars for "A" with any values that need to be returned, then cause "A" to be reactivated. Script "A" will then be able to resume from where it left off, but now it will also have any values that were returned from script "B".

Subject: Re: Maintaining state
Posted by rafs on Thu, 27 Aug 2015 19:11:14 GMT
View Forum Message <> Reply to Message

Yes, this is good. Thank you. I am "watching" variables: $_POST, $fieldarray, $script_vars to see what is happening.