

## Re: passing database data to a sub

---

*Source:* <http://coding.derkeiler.com/Archive/Perl/perl.beginners/2005-04/msg00460.html>

---

- *From:* [security.department@xxxxxxxxx](mailto:security.department@xxxxxxxxx) (John Doe)
  - *Date:* Mon, 18 Apr 2005 09:17:45 +0200
- 

Hi Mick

Just some additional comments to Charles':

Am Montag, 18. April 2005 06.36 schrieb Hawkes, Mick I:

```
> Wow!
> Thanks Charles, it will take a while to digest all that.
> Some questions though...
>
> You said:
>
>
> Second, GetOfficers() is called as a subroutine, not as a method of an
> object. But it is written as an object method. Try this. $template->param(
> db_loop => $self->GetOfficers() );
>
> I'm not sure of the difference, why isn't it a subroutine? I not completely
> sure about this 'shift' thing anyway :-)
>
> Also, I put all the changes in you recommended, but I still get the same
> error!
>
> Error executing run mode 'Mode_0': can't call method "param" on an
> undefined value at Test_code.pm line 283
>
>
> Here is all the code, this is just some test code, ok, the doco isnt
> correct.
>
> package Test_code;
> use base 'CGI::Application';
> use HTML::Template;
> use Data::FormValidator;
> use DBI;
> use DBI qw(:sql_types);
> use strict;
>
>
> #####
```

Re: passing database data to a sub

```
> # Subroutine Name: setup
> # Purpose: To set up the start mode and the various run modes...
> # It also sets up the database.
> # 18-Nov-03 J.Bews :- Initial coding.
> # 23-Nov-03 J.Bews :- Changed location for the Templates. Now in the same
> # location as the HTML files.
> # 02-Dec-03 J.Bews :- Converted to OCL_Database.
> # 22-Dec-03 J.Bews :- Deleted mode_5. Incorporated into mode_4.
> # 14-May-04 J.Bews :- Realised that I was still using the old database
> # in the c:\Program Files\Abyss\cgi-bin\OCL_Database subdirectory. Changed
> # over to the C:\xitami\cgi-bin\OCL_Database subdirectory.
> #####
> sub setup
> {
> my $self = shift;
> # Set the start mode.
> $self->start_mode('mode_0');
> # Set the run modes.
> $self->run_modes
> (
> 'mode_0'=>'mainmenu',
> 'mode_1'=>'insertproject',
> );
>
> # Connect to the database using DBI and ODBC.
> # Open connection to an Access 97 database called 'Sample_db.mdb'.
> my $file = "c:\\xitami\\cgi-bin\\ToolBox\\OCL\\OCL.mdb";
```

(Rather a detail:)

No need to use double quotes with double-backslashes. Simply use

```
my $file = 'c:\xitami\cgi-bin\ToolBox\OCL\OCL.mdb';
```

The same holds on a line below.

```
> my $driver = "driver={ Microsoft Access Driver (*.mdb) }; DBQ=$file";
> $self->param('mydbh' => DBI->connect("dbi:ODBC:$driver"));
> my $dbh = $self->param('mydbh');
> $dbh->{'LongReadLen'}= 5000; # expand the size of the Memo read for
> Access 97
>
>
> # Path to use for Templates....
> # $self->tmpl_path('./'); # ie the same directory as this script.
> $self->tmpl_path("c:\\xitami\\webpages\\TestBed\\");
> }
>
> #####
> # Subroutine Name: teardown
> # Purpose: To disconnect the database when we are all done.
```

## Re: passing database data to a sub

```
> # 18-Nov-03 J.Bews Initial coding.... the disconnect() code does not seem
> to # work for ActiveState Perl Ver 5.6 or Ver 5.8 so I have simply left it
> out.
> #####
> sub teardown
> {
> my $self = shift;
> $self->param('mydbh')->disconnect(); #this line causes problems....
```

It causes problems because you use a string as an object (and try to get a param with the name of the result of the method call).

```
$self->param('mydbh')->disconnect();
```

could solve it, although this short version of getting an object by param() and instantly calling it is generally not recommended because you can't test if there is an object before calling one of its methods.

```
> }
>
>
> #####
> # Subroutine Name: mainmenu
> # Run Mode: rm=mode_0
> # HTML Template: 'ocl_mainmenu.tmpl.htm'
> # Purpose: To show a main main menu.
> # Note...this subroutine uses the template 'db_mainmenu.tmpl.htm' to give
> # the user a list of options.
> # Basically it returns the NAME attribute='rm' with a VALUE attribute =
> # 'mode_1', 'mode_2', etc. depending on the option selected.
> #
> # 29-Nov-03 J.Bews Initial coding.
> # 02-Dec-03 J.bews Recoded for OCL... ie ocl_mainmenu.tmpl.htm
> #
> # To do list:
> # 1)
> #####
> sub mainmenu
> {
> my $self = shift;
> my $dbh = $self->param('mydbh');
>
> # Get the CGI query object so that you can use the CGI.pm modules.
> my $q = $self->query();
> # Setup the template to use for the output.
> my $template = $self->load_tmpl('test2.tmpl.htm');
>
> # call param to fill in the loop with the loop data by reference.
> $template->param(db_loop => $self>GetOfficers());
> # Output the template...
> $template->output;
```

Re: passing database data to a sub

Re: passing database data to a sub

```
> }
>
> #####
>
>
> #####
> sub insertproject
> {
> my $self = shift; # get the passed parameters.
> my $q = $self->query(); # get acopy of the CGI object.
> my $dbh = $self->param('mydbh'); # get the database handle.
>
> ## Construct the SQL Statement
> # Get the values from the form.
> # my $HTML_OCLRef = $q->param("HTML_OCLRef");
> # my $HTML_ProjectID = $q->param("HTML_ProjectID");
> my $HTML_Priority=$q->param("HTML_Priority")||undef;
> my $HTML_Status=$q->param("HTML_Status");
> my $HTML_Officer=$q->param("HTML_Officer")||undef;
>
>
> # Setup the template to use for the output.
> my $template = $self->load_tmpl('test2.tmpl.htm');
> #set up the data to give to the HTML template....
> $template->param(HTML_Priority => $HTML_Priority);
> $template->param(HTML_Status => $HTML_Status);
> $template->param(HTML_Officer => $HTML_Officer);
>
> # Output the template...
> $template->output;
>
> }
>
> sub GetOfficers {
> my $self = shift;
> my $sql = 'SELECT Name as HTML_ProjectName FROM qryOfficer';
> my $dbh = $self->param( 'mydbh' );
> my $sth = $dbh->prepare( $sql );
> $sth->execute()
```

Senseless because you don't catch the results of this call to a variable from witch you can get the select results.

```
> || die qq(Could not execute SQL: "$sql");
```

(Rather a detail:)

Better use 'or' instead of '||' because it's lower precedence compared with '='. In the case of using die, there is no visible difference, but basically, in

Re: passing database data to a sub

`$var = $a || $b; # $a and $b being anything assignable`

\$b is assigned to \$var if \$a results in a false value. Using brackets to show the precedence:

`$var=($a || $b)`  
`($var=$a) or ($b)`

`> return fetchall_arrayref( { HTML_ProjectName => 1 } );`

Ok, I never used CGI::Application and HTML::Template, but where does this sub come from?

There is such a method for the result of \$sth->execute, and called with other arguments.

```
> }
>
>
> #####
> # Perl Packages need to return TRUE so that some of the default functions
> # (ie 'require' and 'use' work. Typically do this by putting a 1; at the
> # end of the package.
> #####
> 1;
>
> # Das Ende Ist Ere!!!!
```

A typo? (Ehre)

Generally, if you are not shure if some code does what you want, just put a print statement after it to see. Charles demonstrated it by using [Data::Dumper](#) for a complex object. It's some "poor man's debugging" and very useful :-)

joe

.

- 
- **Follow-Ups:**
    - ◆ [RE: passing database data to a sub](#)  
◇ From: Charles K. Clarkson
  - **References:**
    - ◆ [RE: passing database data to a sub](#)  
◇ From: Mick I Hawkes
  - Prev by Date: [last question on the switch statement ....](#)
  - Next by Date: [Number of Items in Array](#)

Re: passing database data to a sub

- Previous by thread: ***RE: passing database data to a sub***
- Next by thread: ***RE: passing database data to a sub***
- Index(es):
  - ◆ ***Date***
  - ◆ ***Thread***