PHP - Where is the best place to initiate a database class?

Asked 16 years ago Modified 13 years ago Viewed 479 times

Part of PHP Collective

0

I recently took my Db initiating code out of the __construct of my Page class and placed it just after I initiate the Page class. I removed it from within the Page class because I want to be able to access it from anywhere (other classes for example). It also takes server, username, password and database arguments to it when initiated, and I don't wish to enter these every time.

Is there a way I can access it from under the Page class now? I've tried a few methods, even global (which I have been told is an awful way to do things) and so far no avail. I am still new to OO, but I am teaching myself as best as I can.

Should I make it a static class? Will this affect the lazy connector to the Db I have setup?

Any help would be much appreciated.

Thank you

[EDIT]

Similar Question: Global or Singleton for database connection?

PHP php oop

Share

Improve this question

Follow

edited May 23, 2017 at 12:10

Community Bot
1 ● 1

asked Dec 18, 2008 at 4:34

alex
490k ● 204 ● 889 ● 990

I asked a similar question Global or Singleton for database connection? – Imran Dec 18, 2008 at 4:41

3 Answers

Sorted by: Highest score (default)

2

A global of some sort (Be that global variables, singleton or some other variant) is an improvement over your previous approach, and as such you're on the right track. Generally speaking though, you should try to minimise the scope of program state (For a number of reasons, which I won't get into here). Having a global variable is in conflict with this principle. There are different solutions to this problem, but the most powerful and often overlooked approach, is to use inversion of control; Instead of obtaining a dependency, your class should receive it. For example, let's say you currently have this

```
class EditUserController {
  function saveUser() {
    $db = Database::GetInstance();
    $db->execute("update users set ...", ...);
  }
}
```

You could change this into:

```
class EditUserController {
  function saveUser($db) {
    $db->execute("update users set ...", ...);
  }
}
```

Passing dependencies on the function-parameter level can be a bit unwieldy though, so a compromise could be to pass it on a per-object level:

```
class EditUserController {
  protected $db;
  function __construct($db) {
    $this->db = $db;
  }
  function saveUser() {
    $this->db->execute("update users set ...", ...);
  }
}
```

This is a fairly common pattern in OO programming. In addition to being more practical than passing in function parameters, it has the additional benefit of separating construction (Where shared dependencies are wired up to each other), from runtime (Where they are used). This makes a lot of things simpler.

Share Improve this answer Follow

answered Dec 18, 2008 at 9:33

troelskn
117k ● 27 ● 134 ● 156

I see a verb in your class name. :(– DanMan May 28, 2012 at 14:55

0

Global variables do have a use, and this would be one of them. Unless it's likely that you're going to be needing multiple database connections, (or even still), then I don't see a problem with setting up a global $db object.

An alternative way is to have a static "Factory" class which you can use to get the object. In Joomla 1.5, the way you access the DB object is like this:

```
$db =& JFactory::getDBO();
```

the getDBO function checks if the DB object has been created: if it has, return a reference to it, otherwise connect and initialise, and then return it.

This could equally apply to other "could-be-made-global" objects, like the current User object.

Share Improve this answer Follow

answered Dec 18, 2008 at 4:38

nickf
546k ● 198 ● 658 ● 725

I'll have a crack at developing something like this. I have seen the =& a few times but I'm still not sure what it means. – alex Dec 18, 2008 at 4:41

it sets the variable to be a *reference*. More info here : php.net/manual/en/language.references.return.php – nickf Dec 18, 2008 at 4:45

I prefer a space between = and &, which doesn't make it look like a new operator. & coming after = makes it look more confusing (as it's not in the same order as +=, -=, *= or /=) – Imran Dec 18, 2008 at 4:51

yeah - i've had "discussions" with fellow coders about that. Apparently doing "= &" is 'more correct', but i personally prefer "=&" – nickf Dec 18, 2008 at 5:06

Unless you're using php ≤ 4, assigning objects by reference is meaningless. It will happen implicitly anyway. – troelskn Dec 18, 2008 at 9:11

0

The singleton method was created to make sure there was only one instance of any class. But, because people use it as a way to shortcut globalizing, it becomes known as lazy and/or bad programming.

Share Improve this answer Follow

answered Dec 13, 2011 at 21:57

user1093284