PHP includes vs OOP

Asked 16 years, 4 months ago Modified 8 years, 7 months ago

Viewed 3k times Part of PHP Collective

11

I would like to have a reference for the pros and cons of using include **files vs objects(classes)** when developing PHP applications.

I know I would benefit from having one place to go for this answer...I have a few opinions of my own but I look forward to hearing others.

A Simple Example:

Certain pages on my site are only accessible to logged in users. I have two options for implementation (there are others but let's limit it to these two)

1. *Create an authenticate.php file and include it on every page. It holds the logic for authentication.*
2. *Create a user object, which has an authenticate function, reference the object for authentication on every page.*

Edit I'd like to see some way weigh the benefits of one over the other. My current (and weak reasons) follow:

Includes - Sometimes a function is just easier/shorter/faster to call Objects - Grouping of

functionality and properties leads for longer term maintenance.

Includes - Less code to write (no constructor, no class syntax) call me lazy but this is true.

Objects - Force formality and a single approach to functions and creation.

Includes - Easier for a novice to deal with Objects - Harder for novices, but frowned upon by professionals.

I look at these factors at the start of a project to decide if I want to do includes or objects. Those are a few pros and cons off the top of my head.

PHP php coding-style

Share

Improve this question

Follow

edited May 18, 2016 at 13:09

Hamza Zafeer

2,438 13 35 47

asked Aug 22, 2008 at 14:41

Markus

1,561 1 19 22

6 Answers

Sorted by: Highest score (default)

These are not really opposite choices. You will have to include the checking code anyway. I read your question

13

as procedural programming vs. OO programming.

Writing a few lines of code, or a function, and including it in your page header was how things were done in PHP3 or PHP4. It's simple, it works (that's how we did it in osCommerce, for example, an eCommerce PHP application).

But it's not easy to maintain and modify, as many developers can confirm.

In PHP5 you'd write a user object which will carry its own data and methods for authentication. Your code will be clearer and easier to maintain as everything having to do with users and authentication will be concentrated in a single place.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:57

Christian Lescuyer

19.2k ● 6 ● 50 ● 45

5

While the question touches on a couple of very debatable issues (OOP, User authentication) I'll skip by those and second Konrad's comment about __autoload. Anyone who knows C/C++ knows how much of a pain including files can be. With autoload, a PHP5 addition, if you choose to use OOP (which I do almost exclusively) you only need use some standard file naming convention and (I would recommend) restricting a single class per file and PHP will do the rest for you. Cleans up the code and you no longer have to worry about remembering to remove

includes that are no longer necessary (one of the many problems with includes).

Share Improve this answer

Follow

answered Aug 22, 2008 at 15:03

Karim

18.6k ● 14 ● 63 ● 71

1

I don't have much PHP experience, although I'm using it at my current job. In general, I find that larger systems benefit from the readability and understandability that OO provides. But things like consistency (don't mix OO and non-OO) and your personal preferences (although only really on personal projects) are also important.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:43

Thomas Owens

116k ● 99 ● 317 ● 436

1

I've learned never to use `include` in PHP except inside the core libraries that I use and one central `include` of these libraries (+ config) in the application. Everything else is handled by a global `__autoload` handler that can be configured to recognize the different classes needed. This can be done easily using appropriate naming conventions for the classes.

This is not only flexible but also quite efficient and keeps the architecture clean.

Share Improve this answer

answered Aug 22, 2008 at 14:49

Follow

Konrad Rudolph
545k ● 138 ● 956 ● 1.2k

0

Can you be a bit more specific? For the example you give you need to use include in both ways. In case 1 you only include a file, in case 2 you need to include the class file (for instance user.class.php) to allow instantiation of the User class.

It depends how the rest of the application is built, is it OO? Use OO.

Share Improve this answer

Follow

answered Aug 22, 2008 at 14:46

Huppie
11.4k ● 4 ● 33 ● 34

0

Whether you do it in classes or in a more procedural style, you simply need to check to ensure that:

1. There is a session;
2. That the session is valid; and,
3. That the user in possession of the session has proper privileges.

You can encapsulate all three steps into one function (or a static method in a Session class might work). Try this:

```
class Session
{
  const GUEST = 0;
  const SUBSCRIBER = 1;
```

```
  const ADMINISTRATOR = 2;

  public static function Type()
  {
    session_start();

    // Depending on how you use sessions on
    // your site, you might just check for the
    // existence of PHPSESSID. If you track
    // every visitor with sessions, however, you
    // might want to assign some separate unique
    // number (that you can track in a DB) to
    // authenticated sessions
    if(!$_SESSION['uniqid'])
    {
      return Session::GUEST;
    }
    else
    {
      // For the best security, don't store the
      // user's access permissions in the $_SESSION,
      // but rather check against the DB. This will
      // ensure that recently deleted or downgraded
      // administrators will not be able to make use
      // of a previous session.

      return THE_ACCESS_LEVEL_ACCORDING_TO_THE_DB
    }
  }
}


// In your files that need to check for authentication
// could also do this in a controller if you're going

if(!(Session::Type() == Session::ADMINISTRATOR))
{
  // Redirect them to wherever you want them to go ins
  // like a log in page or something like that.
}
```

Share Improve this answer answered Aug 22, 2008 at 14:56

Follow

Brian Warshaw

23k ● 9 ● 54 ● 72