

Make your own CMS system

By using the FileSystem class, create your own CMS system. This CMS should allow the user to update and modify content by changing the content stored in files. Use an Object Oriented approach to coding it, where you write the business logic and simply use the interface of the class.

Process

You must build this CMS based upon understanding how to use the FileSystem class and then apply it. You are encouraged to work with the process yourself where you apply your own understanding of the FileSystem class.

However a general process can be described as:

- 1) Set up a new project (i.e. new folder in your web root, maybe use the boilerplate)
- 2) Set up a structure like the one below (where "content" is a subfolder)
- 3) Unpack the filesystem example in another folder
- 4) Examine it, look at how the interface is used (and see below here)
- 5) Copy the filesystem.class.php to own folder
- 6) Use it on the index.php to show the content
- 7) Look at the **edit** example to see how the admin could work (in particular process.php)
- 8) On the admin.php page make one <textarea> for the first piece of content and load in the contents of that file
- 9) Make the form go to process.php via the POST method
- 10) Make process.php save the files and redirect to admin
- 11) Repeat 8-10 for the other content files

Structure

The CMS should have the following structure:

- index.php - The front of the site that normal users will see
- admin.php - The admin page that users can access to modify content
- content/news.html - News content, formatted as HTML
- content/contact.html - Contact information that is used in the footer
- content/campaign.html - Should contain the html code for a campaign banner

Filesystem Code

You are to use the FileSystem class (named "filesystem.class.php") as foundation for the CMS.

Hint: You create a new instance of this class by `$file = new FileSystem;` Of course you will need to include it first `include("filesystem.class.php");`.

This class has the following interface:

`read($file)` - Reads the file and returns the contents of the file

`write($file,$content)` - Writes the \$content to the \$file

`isEmpty($file)` - Checks if \$file is empty and returns true or false

index.php

This page should display the content from:

content/news.html

content/contact.html

content/campaign.html (should display this image)

To include this content you must use the `$file->read()` function (and not `include();`)

admin.php

This page should provide the user access to modifying the content of the news/contact/campaign files. Use the `$file->read()` and `$file->write()`

Hint: For the campaign banner, you can use an online placeholder (e.g. <https://placeholder.it>) like this:

```

```