phphooks_class doc

Contents

Package phphooks Procedural Elements

phphooks.class.php

- **Package** phphooks
- **Author** eric.wzy@gmail.com
- **Version** 1.0

PLUGINS_FOLDER = 'plugins/' *[line 11]*

Package phphooks Classes

Class phphooks

[line 13]

- **Package** phphooks

phphooks::$hooks

mixed = array() *[line 25]*

hooks data

phphooks::$plugins

mixed = array() *[line 19]*

plugins data

void function phphooks::add_hook($tag, $function, [$priority = 10]) *[line 112]*

Function Parameters:

- *string* **$tag** The name of the hook.
- *string* **$function** The function you wish to be called.
- *int* **$priority** optional. Used to specify the order in which the functions associated with a particular action are executed.(range 0~20, 0 first call, 20 last call)

attach custom function to hook

- **Since** 1.0

option. function phphooks::execute_hook($tag, [$args = '']) *[line 141]*
Function Parameters:

- *string* **$tag** The name of the hook.
- *array* **$args** optional.The arguments the function accept (default none)

execute all functions which are attached to hook, you can provide argument (or arguments via array)

- **Since** 1.0

array. function phphooks::filter_hook($tag, [$args = '']) *[line 168]*
Function Parameters:

- *string* **$tag** The name of the hook.
- *array* **$args** optional.The arguments the function accept to filter(default none)

filter $args and after modify, return it. (or arguments via array)

- **Since** 1.0

void function phphooks::hook_exist($tag) *[line 128]*
Function Parameters:

- *string* **$tag** The name of the hook.

check whether any function is attached to hook

- **Since** 1.0

void function phphooks::load_plugins([$from_folder = PLUGINS_FOLDER]) *[line 82]*
Function Parameters:

- *var* **$from_folder** option. load plugins from folder, if no argument is supplied, a 'plugins/' constant will be used

load plugins from specific folder, includes all *.plugin.php files

- **Since** 1.0

void function phphooks::register_plugin($plugin_id, [$data = ''], $plugin_id.) *[line 195]*
Function Parameters:

- *string* **$plugin_id.** The name of the plugin.
- *array* **$data** optional.The data the plugin accessorial(default none)
- **$plugin_id**

register plugin data in $this->plugin

- **Since** 1.0

void function phphooks::set_hook($tag) *[line 34]*
Function Parameters:

- *string* **$tag** The name of the hook.

register hook name/tag, so plugin developers can attach functions to hooks

- **Since** 1.0

void function phphooks::set_hooks($tags, $tag) *[line 45]*
Function Parameters:

- *array* **$tag** The name of the hooks.
- **$tags**

register multiple hooks name/tag

- **Since** 1.0

void function phphooks::unset_hook($tag) *[line 58]*
Function Parameters:

- *string* **$tag** The name of the hook.

write hook off

- **Since** 1.0

void function phphooks::unset_hooks($tags, $tag) *[line 69]*

Function Parameters:

- *array* **$tag** The name of the hooks.
- **$tags**

write multiple hooks off

- **Since** 1.0

Package default Procedural Elements

DebugDemo.php

- **Package** default

void function display_workers() *[line 41]*

Displays a table of the workers

string function row_color($i) *[line 24]*

Function Parameters:

- *int* **$i**

Returns 'white' for even numbers and 'yellow' for odd numbers

Appendices

Appendix A - Class Trees

Package default

Package phphooks

phphooks

- phphooks

Index