phphooks_class doc

Contents

Package phphooks Procedural Elements . . . 2
phphooks.class.php . . . 2
Define PLUGINS_FOLDER . . . 2
Package phphooks Classes . . . 3
Class phphooks . . . 3
Var $active_plugins . . . 3
Var $hooks . . . 3
Var $plugins . . . 3
Var $plugins_header . . . 3
Method add_hook . . . 4
Method execute_hook . . . 4
Method filter_hook . . . 4
Method get_plugins_header . . . 5
Method hook_exist . . . 5
Method load_plugins . . . 5
Method register_plugin . . . 6
Method set_hook . . . 6
Method set_hooks . . . 7
Method unset_hook . . . 7
Method unset_hooks . . . 7
Package default Procedural Elements . . . 10
DebugDemo.php . . . 10
Function display_workers . . . 10
Function row_color . . . 10
Appendices . . . 11
Appendix A - Class Trees . . . 12
default . . . 12
phphooks . . . 12

Package phphooks Procedural Elements

phphooks.class.php

- **Package** phphooks
- **Author** eric.wzy@gmail.com
- **Version** 1.1
- **License** LGPL

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

Package phphooks Classes

Class phphooks
[line 13]

- **Package** phphooks

phphooks::$active_plugins

array = NULL *[line 25]*

UNSET means load all plugins, which is stored in the plugin folder. ISSET just load the plugins in this array.

phphooks::$hooks

array = array () *[line 37]*

hooks data

phphooks::$plugins

array = array () *[line 19]*

plugins option data

phphooks::$plugins_header

array = array () *[line 31]*

all plugins header information array.

void function phphooks::add_hook($tag, $function, [$priority = 10], $tag., $function.) *[line 169]*

Function Parameters:

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

attach custom function to hook

- **Since** 1.0

optional. function phphooks::execute_hook($tag, [$args = ''], $tag.) *[line 197]*

Function Parameters:

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

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 = ''], $tag.) *[line 223]*

Function Parameters:

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

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

- **Since** 1.0

array. function phphooks::get_plugins_header([$from_folder = PLUGINS_FOLDER]) *[line 122]*

Function Parameters:

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

return the all plugins ,which is stored in the plugin folder, header information.

- **Since** 1.1

void function phphooks::hook_exist($tag) *[line 184]*

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 94]*

Function Parameters:

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

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

- **Since** 1.0

void function phphooks::register_plugin($plugin_id, [$data = ''], $plugin_id.) *[line 249]*

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, $tag.) *[line 46]*

Function Parameters:

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

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

- **Since** 1.0

void function phphooks::set_hooks($tags, $tags.) *[line 57]*
Function Parameters:

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

register multiple hooks name/tag

- **Since** 1.0

void function phphooks::unset_hook($tag, $tag.) *[line 70]*
Function Parameters:

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

write hook off

- **Since** 1.0

void function phphooks::unset_hooks($tags, $tags.) *[line 81]*
Function Parameters:

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

D

display_workers() . . . 10
DebugDemo.php . . . 10

P

phphooks::load_plugins() . . . 5
*load plugins from specific folder, includes *.plugin.php files*
phphooks::hook_exist() . . . 5
check whether any function is attached to hook
phphooks::get_plugins_header() . . . 5
return the all plugins ,which is stored in the plugin folder, header information.
phphooks::register_plugin() . . . 6
register plugin data in $this->plugin
phphooks::set_hook() . . . 6
register hook name/tag, so plugin developers can attach functions to hooks
phphooks::unset_hooks() . . . 7
write multiple hooks off
phphooks::unset_hook() . . . 7
write hook off
phphooks::set_hooks() . . . 7
register multiple hooks name/tag
phphooks::filter_hook() . . . 4
filter $args and after modify, return it. (or arguments via array)
phphooks::execute_hook() . . . 4
execute all functions which are attached to hook, you can provide argument (or arguments via array)
phphooks::$active_plugins . . . 3
UNSET means load all plugins, which is stored in the plugin folder. ISSET just load the plugins in this array.
phphooks . . . 3
PLUGINS_FOLDER . . . 2
phphooks::$hooks . . . 3
hooks data
phphooks::$plugins . . . 3
plugins option data
phphooks::add_hook() . . . 4
attach custom function to hook
phphooks::$plugins_header . . . 3
all plugins header information array.
phphooks.class.php . . . 2

R

row_color() 10