Search

Getting Started
- Introduction
- A simple tutorial

Language Reference
- Basic syntax
- Types
- Variables
- Constants
- Expressions
- Operators
- Control Structures
- Functions
- Classes and Objects
- Namespaces
- Enumerations
- Errors
- Exceptions
- Fibers
- Generators
- Attributes
- References Explained
- Predefined Variables
- Predefined Exceptions
- Predefined Interfaces and Classes
- Predefined Attributes
- Context options and parameters
- Supported Protocols and Wrappers

Security
- Introduction
- General considerations
- Installed as CGI binary
- Installed as an Apache module
- Session Security
- Filesystem Security
- Database Security
- Error Reporting
- User Submitted Data
- Hiding PHP
- Keeping Current

Features
- HTTP authentication with PHP
- Cookies
- Sessions
- Dealing with XForms
- Handling file uploads
- Using remote files
- Connection handling
- Persistent Database Connections
- Command line usage

Garbage Collection
DTrace Dynamic Tracing

Function Reference
Affecting PHP's Behaviour
Audio Formats Manipulation
Authentication Services
Command Line Specific Extensions
Compression and Archive Extensions
Cryptography Extensions
Database Extensions
Date and Time Related Extensions
File System Related Extensions
Human Language and Character Encoding Support
Image Processing and Generation
Mail Related Extensions
Mathematical Extensions
Non-Text MIME Output
Process Control Extensions
Other Basic Extensions
Other Services
Search Engine Extensions
Server Specific Extensions
Session Extensions
Text Processing
Variable and Type Related Extensions
Web Services
Windows Only Extensions
XML Manipulation
GUI Extensions

Keyboard Shortcuts

?
: This help

j
: Next menu item

k
: Previous menu item

g p
: Previous man page

g n
: Next man page

G
: Scroll to bottom

g g
: Scroll to top

g h
: Goto homepage

g s
: Goto search
(current page)

/
: Focus search box

Суперглобальные переменные »
« Неявное использование механизма ссылок

- Руководство по PHP
- Справочник языка

Change language: Russian

Submit a Pull Request Report a Bug

Предопределённые переменные

PHP предоставляет всем скриптам большое количество предопределённых переменных. Эти переменные содержат всё, от внешних данных до переменных среды окружения, от текста сообщений об ошибках до последних полученных заголовков.

Содержание

- Суперглобальные переменные — Встроенные переменные, которые всегда доступны во всех областях
- $GLOBALS — Ссылки на все переменные глобальной области видимости
- $_SERVER — Информация о сервере и среде исполнения
- $_GET — Переменные HTTP GET
- $_POST — Переменные HTTP POST
- $_FILES — Переменные файлов, загруженных по HTTP
- $_REQUEST — Переменные HTTP-запроса
- $_SESSION — Переменные сессии
- $_ENV — Переменные окружения
- $_COOKIE — HTTP Cookies
- $php_errormsg — Предыдущее сообщение об ошибке
- $http_response_header — Заголовки ответов HTTP
- $argc — Количество аргументов, переданных скрипту
- $argv — Массив переданных скрипту аргументов

User Contributed Notes 25 notes

40
New York PHP ¶
18 years ago
```
Warning: $_SERVER['PHP_SELF'] can include arbitrary user input. The documentation should be updated to reflect this.

The request "http://example.com/info.php/attack%20here" will run /info.php, but in Apache $_SERVER['PHP_SELF'] will equal
"/info.php/attack here". This is a feature, but it means that PHP_SELF must be treated as user input.

The attack string could contain urlencoded HTML and JavaScript (cross-site scripting) or it could contain urlencoded
linebreaks (HTTP response-splitting).

The use of $_SERVER['SCRIPT_NAME'] is recommended instead.
```

22
josh,endquote,com ¶
20 years ago
```
Running PHP 4.3 under IIS 5 on Windows XP, there is no $_SERVER['REQUEST_URI'] variable. This seems to fix it:

if(!isset($_SERVER['REQUEST_URI'])) {
$_SERVER['REQUEST_URI'] = substr($_SERVER['argv'][0], strpos($_SERVER['argv'][0], ';') + 1);
}
```

7
Aardvark ¶
17 years ago
```
$_GET may not handle query string parameter values which include escaped Unicode values resulting from applying the
JavaScript "escape" function to a Unicode string.
To handle this the query parameter value can be obtained using a function such as:

function getQueryParameter ($strParam) {
```

```
$aParamList = explode('&', $_SERVER['QUERY_STRING']);
$i = 0;
while ($i < count($aParamList)) {
$aParam = split('=', $aParamList[$i]);
if ($strParam == $aParam[0]) {
return $aParam[1];
}
}
return "";
}
```

or by directly building an array or query string values and then processing the parameter string using a function such as the "unescape" function which can be found at http://www.kanolife.com/escape/2006/03/unicode-url-escapes-in-php.html (or http://www.kanolife.com/escape/ for related info).

up
down
9

jameslporter at gmail dot com ¶

17 years ago

Refer to CanonicalName if you are not getting the ServerName in the $_SERVER[SERVER_NAME] variable....This was a pain to figure out for me...now it works as expected by turning canonical naming on.

http://www.apacheref.com/ref/http_core/UseCanonicalName.html

up
down
6

daniel at softel dot jp ¶

18 years ago

Note that $php_errormsg may contain a newline character. This can be problematic if you are trying to output it with a JavaScript "alert()" for example.

up
down
7

danvasile at pentest dot ro ¶

16 years ago

If you have problems with $_SERVER['HTTPS'], especially if it returns no values at all you should check the results of phpinfo(). It might not be listed at all.
Here is a solution to check and change, if necessary, to ssl/https that will work in all cases:

```
<?php
if ($_SERVER['SERVER_PORT']!=443) {
$sslport=443; //whatever your ssl port is
$url = "https://". $_SERVER['SERVER_NAME'] . ":" . $sslport . $_SERVER['REQUEST_URI'];
header("Location: $url");
}
?>
```

Of course, this should be done before any html tag or php echo/print.

up
down
9

nathan ¶

17 years ago

Also on using IPs to look up country & city, note that what you get might not be entirely accurate. If their ISP is based in a different city or province/state, the IPs may be owned by the head office, and used across several areas.
You also have rarer situations where they might be SSHed into another server, on the road, at work, at a friend's... It's a nice idea, but as the example code shows, it should only be used to set defaults.

up
down
6

mrnopersonality at yahoo dot com ¶

19 years ago

Nothing about the message-body ...

You can get cookies, session variables, headers, the request-uri , the request method, etc but not the message body. You may want it sometimes when your page is to be requested with the POST method.

Maybe they should have mentioned $HTTP_RAW_POST_DATA or php://stdin

up
down
6

youdontmeanmuch [at] yahoo.com ¶

19 years ago

Be carful when using $_SERVER['DOCUMENT_ROOT']; in your applications where you want to distribute them to other people with different server types. It isnt always supported by the webserver (IIS).

up
down
5

Joe Marty ¶

17 years ago

I think it is very important to note that PHP will automatically replace dots ('.') AND spaces (' ') with underscores ('_') in any incoming POST or GET (or REQUEST) variables.

This page notes the dot replacement, but not the space replacement:
http://us2.php.net/manual/en/language.variables.external.php

The reason is that '.' and ' ' are not valid characters to use in a variable name. This is confusing to many people, because most people use the format $_POST['name'] to access these values. In this case, the name is not used as a variable name but as an array index, in which those characters are valid.

However, if the register_globals directive is set, these names must be used as variable names. As of now, PHP converts the names for these variables before inserting them into the external variable arrays, unfortunately - rather than leaving them as they are for the arrays and changing the names only for the variables set by register_globals.

If you want to use:
<input name="title for page3.php" type="text">

The value you will get in your POST array, for isntance would be:
$_POST['title_for_page3_php']

up
down
6

drew dot griffiths at clare dot net ¶

18 years ago

Re: You can take advantage of 404 error to an usable redirection using REQUEST_URI ...

Whilst this is effective, a line in the .htaccess such as:

```
RewriteEngine On
RewriteRule ^profiles/([A-Za-z0-9-]+) showprofile.php?profile=$1 [L,NC,QSA]
```

will throw the requested profile in a variable $profile to the showprofile.php page.

You can further enhance the url (e.g http://servername/profiles/Jerry/homeaddress/index.htm) and the second variable value homeaddress becomes available in $url_array[3] when used below $url_array=explode("/",$_SERVER['REQUEST_URI']);

Hope this helps - Works well for me

Drew

up
down
4

Gregory Boshoff ¶

19 years ago

The Environment variable $ENV is useful for coding portable platform specific application constants.

```
// Define a Windows or else Linux root directory path
$_ENV['OS'] == 'Windows_NT' ? $path = 'L:\\www\\' : $path = ' /var/www/';

define('PATH', $path);

echo PATH;
```

up
down
3

Ben XO ¶

17 years ago

So you have an application in your web space, with a URL such as this:

http://<host>/<installation_path>/

and pages such as

http://<host>/<installation_path>/subfolder1/subfolder2/page.php

You have a file called config.php in <installation_path> which is include()d by all pages (in subfolders or not).

How to work out <installation_path> without hard-coding it into a config file?

```
<?php

// this is config.php, and it is in <installation_path>
// it is included by <installation_path>/page.php
// it is included by <installation_path>/subfolder/page2.php
// etc

$_REAL_SCRIPT_DIR = realpath(dirname($_SERVER['SCRIPT_FILENAME'])); // filesystem path of this page's directory (page.php)
$_REAL_BASE_DIR = realpath(dirname(__FILE__)); // filesystem path of this file's directory (config.php)
$_MY_PATH_PART = substr( $_REAL_SCRIPT_DIR, strlen($_REAL_BASE_DIR)); // just the subfolder part between
<installation_path> and the page

$INSTALLATION_PATH = $_MY_PATH_PART
? substr( dirname($_SERVER['SCRIPT_NAME']), 0, -strlen($_MY_PATH_PART) )
: dirname($_SERVER['SCRIPT_NAME'])
; // we subtract the subfolder part from the end of <installation_path>, leaving us with just <installation_path> :)

?>
```

up
down
2

dusted at dusted dot dk ¶

12 years ago

I use HTTP_X_FORWARDED_FOR because my webserver is behind a reverse proxy.
This can be made secure:
Configure the reverse proxy to block this field, and override it correctly.
Configure the apache server to only accept incoming connections from the reverse proxy.

up
down
2

php-net dot ucn dot extranet dot sys at dark-chiaki dot net ¶

16 years ago

In addition to mfyahya at gmail dot com (2007-06-07 03:33):

If You are working with the Apache module mod_rewrite and want to set some environment vars, the Apache manual says this vars could be accessed in CGI using $ENV{VAR}. In PHP You might want to write $_ENV['VAR'] to get the value of VAR, but You have to access if via $_SERVER, and in some different ways:

```
1. Example: .htaccess and example.php

RewriteEngine on
RewriteRule ^?var1=([^;]*);var2=([^;]*)$ \
- [E=VAR1:$1,E=VAR2:$2]

<?php echo($_SERVER['VAR1']."\r\n"
.$_SERVER['VAR2']); ?>

2. Example: .htaccess and index.php

RewriteEngine on
RewriteRule ^index\.php$ - [L]
RewriteRule ?var1=([^;]*);var2=([^;]*)$ \
index.php [E=VAR1:$1,E=VAR2:$2]

<?php echo($_SERVER['REDIRECT_VAR1']."\r\n"
.$_SERVER['REDIRECT_VAR2']); ?>

Note: If any RewriteRule matches, an internal redirect than restarts (after the last defined rule, or immediately after
the matched rule having a L-flag) checking the entire rule set again. For an internal redirect every defined VAR gets an
'REDIRECT_' prefix, i.e. VAR1 will be REDIRECT_VAR1, VAR2 will be REDIRECT_VAR2.

Of course, You can (additionally) redefine the original VAR:

RewriteEngine on
RewriteRule ^index\.php$ \
- [E=VAR1:%{REDIRECT_VAR1},E=VAR2:%{REDIRECT_VAR2},L]
RewriteRule ?var1=([^;]*);var2=([^;]*)$ \
index.php [E=VAR1:$1,E=VAR2:$2]

With this, You will have $_SERVER['REDIRECT_VAR*'] -and- $_SERVER['VAR*'].

***

The given examples are only for explanation, in any case they are not intended to fit Your needs. The "\<CRLF><SP>" in the
.htaccess examples are only for display purpose, they should not occur in a real .htaccess file. The argument separator
';' in links can also be '&', but this may cause some trouble with HTML/XHTML. See the following pages for more
information about this issue:
- http://www.w3.org/TR/html4/appendix/notes.html#h-B.2.2
- http://www.w3.org/QA/2005/04/php-session
```

up
down
2

Gregory Boshoff ¶

18 years ago

```
$_SERVER['QUERY_STRING']

Does not contain XHTML 1.1 compliant ampersands i.e. &amp;

So you will need to do something like this if you are to use $_SERVER['QUERY_STRING'] in URL's.

// XHTML 1.1 compliant ampersands
$_SERVER['QUERY_STRING'] =
str_replace(array('&amp;', '&'), array('&', '&amp;'),
$_SERVER['QUERY_STRING']);
```

up
down
2

Nicolae Namolovan ¶

16 years ago

SECURITY RISK !

Never ever trust the values that comes from $_SERVER.

HTTP_X_FORWARDED, HTTP_X_FORWARDED_FOR, HTTP_FORWARDED_FOR, HTTP_FORWARDED, etc.. can be spoofed !

To get the ip of user, use only $_SERVER['REMOTE_ADDR'], otherwise the 'ip' of user can be easily changed by sending a HTTP_X_* header, so user can escape a ban or spoof a trusted ip.

Of course this is well know, but I don't see it mentioned in these notes..

If you use the ip only for tracking (not for any security features like banning or allow access to something by ip), you can also use HTTP_X_FORWARDED to get user's ip what are behind proxy.

up
down
1

jsan ¶

16 years ago

@White-Gandalf: one can control this behavior by setting:

UseCanonicalName On|Off

in their apache config (at least, on *ix platforms).

On => $_SERVER['SERVER_NAME'] is the ServerName var from either the global server or virtual host, whichever wraps the PHP app closest.

Off => Whatever was in the Host: header sent by the client.

up
down
1

Anonymous ¶

17 years ago

I was unable to convince my hosting company to change their installation of PHP and therefore had to find my own way to computer $_SERVER["DOCUMENT_ROOT"]. I eventually settled on the following, which is a combination of earlier notes (with some typos corrected):

```
<?php
if ( ! isset($_SERVER['DOCUMENT_ROOT'] ) )
$_SERVER['DOCUMENT_ROOT'] = str_replace( '\\', '/', substr(
$_SERVER['SCRIPT_FILENAME'], 0, 0-strlen($_SERVER['PHP_SELF']) ) );
?>
```

up
down
0

autofei at gmail dot com ¶

13 years ago

if you try to run php through command line, for example: php.exe c:\AppServ\www\cron_cache.php. You better avoid to use $_SERVER['DOCUMENT_ROOT'], because it will return nothing. Instead, you can use dirname(__FILE__). The reason to use command line running php is set it as Windows Scheduled Tasks. I did not test under Linux environment, but might be same.

up
down
1

mfyahya at gmail dot com ¶

18 years ago

If you use Apache's redirection features for custom error pages or whatever, the following Apache's REDIRECT variables are also available in $_SERVER:

$_SERVER['REDIRECT_UNIQUE_ID]'
$_SERVER['REDIRECT_SCRIPT_URL]'
$_SERVER['REDIRECT_SCRIPT_URI]'
$_SERVER['REDIRECT_SITE_ROOT]'
$_SERVER['REDIRECT_SITE_HTMLROOT]'

```
$_SERVER['REDIRECT_SITE_CGIROOT]'
$_SERVER['REDIRECT_STATUS]'
$_SERVER['REDIRECT_QUERY_STRING]'
$_SERVER['REDIRECT_URL]'
```

I'm not sure if this is a complete list though

up
down
0

spamtrap at scottydm dot com ¶

16 years ago

I was a little frustrated by the fact that some of the _SERVER variables didn't seem to exist, so I did a bit of Googling and found the answer: many of these variables are supplied by the web server and not all web servers supply the same set of variables.

I found a comparison between Apache v1.3.29 and IIS v5.1 on this page: http://koivi.com/apache-iis-php-server-array.php Useful for those of us doing cross-platform development.

While running experiments with different browsers I noticed some of the HTTP_* variables come and go depending on the browser used, or in the case of Opera by diddling the "user mode" (the widget that lets you look at a page as text only, etc.). For example: in IE and Opera HTTP_KEEP_ALIVE was missing, but was present in Firefox and Mozilla, and when I fiddled with Opera's "user mode" I got somethings called HTTP_TE and HTTP_CACHE_CONTROL.

So, what you get is dependent on the web server AND the browser.

I did see one IIS supplied variable not on that list: REQUEST_TIME, which seems to be in Unix timestamp format.

While researching this I discovered there are plenty of people who have their phpinfo() page visible and indexed on a few search engines. For those who want to dig a bit deeper than that nice web page comparing Apache to IIS, looking at other peoples' phpinfo() pages could be useful. You get the version of PHP plus OS and web server they use, along with all the _SERVER variables. I found the highest percent of signal-to-noise by searching for "phpinfo()" (with the quotes) on Dogpile: http://www.dogpile.com/

up
down
-2

todd dot kisov at yahoo dot com ¶

17 years ago

To convert query string parameter values ($_GET, $_REQUEST), which include escaped Unicode values resulting from applying the JavaScript "escape" function to a Unicode string (%uNNNN%uNNNN%uNNNN) fast and simple is to use PECL JSON extension:

```
function JavaScript_Unicode_URL_2_Str($js_uni_str) {
$res = preg_replace('/%u([[:alnum:]]{4})/', '\\u\1', $js_uni_str);
$res = str_replace('"', '\"', $res); // if in str "
$res = json_decode('["'.$res.'"]'); // JavaScrip array with string element
$res = $res[0];
$res = iconv('UTF-8', ini_get('default_charset'), $res);
return $res;
}
```

up
down
-2

marcus at lastcraft dot com ¶

19 years ago

The variable $php_errormsg is not populated if you have XDebug running.

up
down
-4

justin dot (nospam)george at gmail dot com ¶

17 years ago

Note that it's a very, very bad idea to append to global variables in a loop, unless you really, really mean to do so in a global context. I just a while ago hung my server with a snippet of code like this:

```
<?php
$host = $_SERVER['HTTP_HOST'];
$uri = rtrim($_SERVER['PHP_SELF'], "/\\");


$GLOBALS['SITE_ROOT'] = "http://$host$uri";

while ($i < somenumber)
readfile($GLOBALS['SITE_ROOT'] = $GLOBALS['SITE_ROOT'] . '/this/file.php');
$i++
}
?>
```

While it is an entertaining and unusual method of creating very long URLs and breaking servers, it's a pretty awesomely bad idea

(Especially considering that the script in question ran concurrently with others of it's type, so the value in $GLOBALS['SITE_ROOT'] was unknown.)

- Справочник языка
  - Основы синтаксиса
  - Типы
  - Переменные
  - Константы
  - Выражения
  - Операторы
  - Управляющие конструкции
  - Функции
  - Классы и объекты
  - Пространства имён
  - Перечисления
  - Ошибки
  - Исключения
  - Fibers
  - Генераторы
  - Атрибуты
  - Объяснение ссылок
  - Предопределённые переменные
  - Предопределённые исключения
  - Встроенные интерфейсы и классы
  - Предопределённые атрибуты
  - Контекстные опции и параметры
  - Поддерживаемые протоколы и обёртки