PHP 7.1.0 Released

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

file_get_contents

(PHP 4 >= 4.3.0, PHP 5, PHP 7)

file_get_contents — Reads entire file into a string

Description ¶

string **file_get_contents** (string `$filename` [, bool `$use_include_path` = false [, resource `$context` [, int `$offset` = 0 [, int `$maxlen`]]]])

This function is similar to file(), except that **file_get_contents()** returns the file in a string, starting at the specified `offset` up to `maxlen` bytes. On failure, **file_get_contents()** will return **`FALSE`**.

file_get_contents() is the preferred way to read the contents of a file into a string. It will use memory mapping techniques if supported by your OS to enhance performance.

> **Note**:
>
> If you're opening a URI with special characters, such as spaces, you need to encode the URI with urlencode().

Parameters ¶

`filename`

> Name of the file to read.

`use_include_path`

> **Note**:
>
> As of PHP 5 the **`FILE_USE_INCLUDE_PATH`** constant can be used to trigger include path search.

`context`

A valid context resource created with stream_context_create(). If you don't need to use a custom context, you can skip this parameter by **NULL**.

`offset`

The offset where the reading starts on the original stream. Negative offsets count from the end of the stream.

Seeking (`offset`) is not supported with remote files. Attempting to seek on non-local files may work with small offsets, but this is unpredictable because it works on the buffered stream.

`maxlen`

Maximum length of data read. The default is to read until end of file is reached. Note that this parameter is applied to the stream processed by the filters.

Return Values ¶

The function returns the read data or **FALSE** on failure.

Warning

This function may return Boolean **FALSE**, but may also return a non-Boolean value which evaluates to **FALSE**. Please read the section on Booleans for more information. Use the === operator for testing the return value of this function.

Errors/Exceptions ¶

An **E_WARNING** level error is generated if `filename` cannot be found, `maxlength` is less than zero, or if seeking to the specified `offset` in the stream fails.

Examples ¶

Example #1 Get and output the source of the homepage of a website

```
<?php
$homepage = file_get_contents('http://www.example.com/');
echo $homepage;
?>
```

Example #2 Searching within the include_path

```
<?php
// <= PHP 5
$file = file_get_contents('./people.txt', true);
// > PHP 5
$file = file_get_contents('./people.txt', FILE_USE_INCLUDE_PATH);
?>
```

Example #3 Reading a section of a file

```php
<?php
// Read 14 characters starting from the 21st character
$section = file_get_contents('./people.txt', NULL, NULL, 20, 14);
var_dump($section);
?>
```

The above example will output something similar to:

```
string(14) "lle Bjori Ro"
```

Example #4 Using stream contexts

```php
<?php
// Create a stream
$opts = array(
  'http'=>array(
    'method'=>"GET",
    'header'=>"Accept-language: en\r\n" .
              "Cookie: foo=bar\r\n"
  )
);

$context = stream_context_create($opts);

// Open the file using the HTTP headers set above
$file = file_get_contents('http://www.example.com/', false, $context);
?>
```

Changelog ¶

Version	Description
7.1.0	Support for negative `offsets` has been added.
5.1.0	Added the `offset` and `maxlen` parameters.

Notes ¶

> **Note**: This function is binary-safe.

Tip

A URL can be used as a filename with this function if the fopen wrappers have been enabled. See fopen() for more details on how to specify the filename. See the Supported Protocols and Wrappers for links to information about what abilities the various wrappers have, notes on their usage, and information on any predefined variables they may provide.

Warning

When using SSL, Microsoft IIS will violate the protocol by closing the connection without sending a *close_notify* indicator. PHP will report this as "SSL: Fatal Protocol Error" when you reach the end of the data. To work around this, the value of error_reporting should be lowered to a level that does not include warnings. PHP can detect buggy IIS server software when you open the stream using the *https://* wrapper and will suppress the warning. When using fsockopen() to create an *ssl://* socket, the developer is responsible for detecting and suppressing this warning.

See Also ¶

- file() - Reads entire file into an array
- fgets() - Gets line from file pointer
- fread() - Binary-safe file read
- readfile() - Outputs a file
- file_put_contents() - Write a string to a file
- stream_get_contents() - Reads remainder of a stream into a string
- stream_context_create() - Creates a stream context
- $http_response_header

add a note

User Contributed Notes 34 notes

up
down
82
Bart Friederichs ¶
4 years ago
file_get_contents can do a POST, create a context for that first:

```
$opts = array('http' =>
  array(
    'method'  => 'POST',
    'header'  => "Content-Type: text/xml\r\n".
      "Authorization: Basic ".base64_encode("$https_user:$https_password")."\r\n",
    'content' => $body,
    'timeout' => 60
  )
);

$context  = stream_context_create($opts);
$url = 'https://'.$https_server;
$result = file_get_contents($url, false, $context, -1, 40000);
```

up
down
28
Stas Trefilov, OpteamIS ¶
5 years ago
here is another (maybe the easiest) way of doing POST http requests from php using its built-in capabilities. feel free to add the headers you need (notably the Host: header) to further customize the request.

note: this method does not allow file uploads. if you want to upload a file with your request you will need to modify the context parameters to provide multipart/form-data encoding (check out http://www.php.net/manual/en/context.http.php) and build the $data_url following the guidelines on http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.2

```
<?php
/**
make an http POST request and return the response content and headers
@param string $url    url of the requested script
```

```
@param array $data    hash array of request variables
@return returns a hash array with response content and headers in the following form:
    array ('content'=>'<html></html>'
        , 'headers'=>array ('HTTP/1.1 200 OK', 'Connection: close', ...)
        )
*/
function http_post ($url, $data)
{
    $data_url = http_build_query ($data);
    $data_len = strlen ($data_url);

    return array ('content'=>file_get_contents ($url, false, stream_context_create (array
('http'=>array ('method'=>'POST'
            , 'header'=>"Connection: close\r\nContent-Length: $data_len\r\n"
            , 'content'=>$data_url
            ))))
        , 'headers'=>$http_response_header
        );
}
?>
```

up
down
10
forestrf at gmail dot com ¶
2 years ago
It is important to write the method in capital letters like "GET" or "POST" and not "get" or "post". Some servers can respond a 400 error if you do not use caps in the method.

up
down
31
joachimb at gmail dot com ¶
8 years ago
Setting the timeout properly without messing with ini values:

```
<?php
$ctx = stream_context_create(array(
    'http' => array(
        'timeout' => 1
        )
    )
);
file_get_contents("http://example.com/", 0, $ctx);
?>
```

up
down
27
colnector bla-at_bla colnect.com ¶
8 years ago
A UTF-8 issue I've encountered is that of reading a URL with a non-UTF-8 encoding that is later displayed improperly since file_get_contents() related to it as UTF-8. This small function should show you how to address this issue:

```
<?php
function file_get_contents_utf8($fn) {
```

```
     $content = file_get_contents($fn);
      return mb_convert_encoding($content, 'UTF-8',
          mb_detect_encoding($content, 'UTF-8, ISO-8859-1', true));
}
?>
```

up
down
15

volkan-k at users dot sourceforge dot net ¶

3 years ago

If you are using file_get_contents() function to retrieve HTTP url and printing HTTP content, you can also send original content-type header using $http_response_header and header() function;

```
<?php

foreach ($http_response_header as $value) {
    if (preg_match('/^Content-Type:/i', $value)) {
        // Successful match
        header($value,false);
    }
}

?>
```

up
down
16

eric at midkotasolutions dot com ¶

5 years ago

At least as of PHP 5.3, file_get_contents no longer uses memory mapping.

See comments on this bug report:

http://bugs.php.net/bug.php?id=52802

up
down
17

francois hill ¶

8 years ago

Seems file looks for the file inside the current working (executing) directory before looking in the include path, even with the FILE_USE_INCLUDE_PATH flag specified.

Same behavior as include actually.

By the way I feel the doc is not entirely clear on the exact order of inclusion (see include). It seems to say the include_path is the first location to be searched, but I have come across at least one case where the directory containing the file including was actually the first to be searched.

Drat.

up
down
9

Yoga Wibowo Aji ¶

5 years ago

read text per line and convert to array

for example, the input file is input.txt
the input file containt text below

one
two
three
four
five

++++++++++++++++++++++++++

read value per line

```
<?php
$data = file_get_contents("input.txt"); //read the file
$convert = explode("\n", $data); //create array separate by new line

for ($i=0;$i<count($convert);$i++)
{
    echo $convert[$i].', '; //write value by index
}
?>
```

++++++++++++++++++++++++++

Output :

one, two, three, four, five,

up
down
7
slymak at gmail dot com ¶
5 years ago

If working file is bigger than 64kb and you getting deadlock. Your buffer is overflow. Here are two way how to avoid that.

1) use temporary file for descriptor

```
<?php
  $descriptorspec = array(
   0 => array("file", "/tmp/ens/a.ens","r"),  // stdin is a pipe that the child will read from
   1 => array("file", "/tmp/ens/a.html","w"),  // stdout is a pipe that the child will write to
   2 => array("file", "/tmp/ens/error-output.txt", "a") // stderr is a file to write to
  );
?>
```

2) inline read using stream_set_blocking. PHP doesn't proper handle last part of file.

```
<?php
  $READ_LEN = 64*1024;
  $MAX_BUF_LEN = 2*$READ_LEN;

  $url = "http://some.domain.com:5984/".$db."/".$member."/contents";
  $src = fopen($url,"r");
```

```
  $cwd = '/tmp';
  $cmd['enscript'] = "/usr/bin/enscript";
  $cmd['enscript-options'] = " -q --language=html --color -Ejcl -o -";
  $descriptorspec = array(
    0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
    1 => array("pipe", "w")   // stdout is a pipe that the child will write to
  );

  $ph=proc_open($cmd['enscript']." ".$cmd['enscript-options'],$descriptorspec,$pipes,$cwd);

  stream_set_blocking($src,0);
  stream_set_blocking($pipes[0],0);
  stream_set_blocking($pipes[1],0);

  $CMD_OUT_OPEN = TRUE; $k = 0;
  while (!feof($pipes[1]) || !feof($src) || $k > 0) {
    if (!feof($src) && $k+$READ_LEN <= $MAX_BUF_LEN) {
      $input .= fread($src,$READ_LEN);
      $k = strlen($input);
    }
    if ($k > 0) {
      $l = fwrite($pipes[0],$input);
      $k -= $l;
      $input = substr($input,$l);
    }
    if ($CMD_OUT_OPEN && $k == 0 && feof($src)) {
      fclose($pipes[0]);
      $CMD_OUT_OPEN = FALSE;
    }
    $output = fread($pipes[1],$READ_LEN);
    $outputn = str_replace("<H1>(stdin)</H1>","",$output);
        echo $outputn;
  }

  fclose($pipes[1]);

  $return_value = proc_close($ph);
?>
```

up
down
5
KrisWebDev ¶
6 years ago

If your file_get_contents freezes during several seconds, here is maybe your answer:

Beware that the default keepalive timeout of Apache 2.0 httpd is 15 seconds. This is true for HTTP/1.1 connections, which is not the default behavior of file_get_contents but you can force it, especially if you are trying to act as a web browser. I don't know if this is also the case for HTTP/1.0 connections.

Forcing the server to close the connection would make you gain those 15 seconds in your script:

```
<?php
$context = stream_context_create(array('http' => array('header'=>'Connection: close')));
```

```
$content = file_get_contents("http://www.example.com/test.html");
?>
```

Another way of resolving slowness issues is to use cURL or fsockopen. Bear in mind that contrary to the behavior of web browsers, file_get_contents doesn't return the result when the web page is fully downloaded (i.e. HTTP payload length = value of the response HTTP "Content-Length" header) but when the TCP connection is closed.
I hope this behavior will change in future releases of PHP.
This has been experienced with PHP 5.3.3.

up
down
6

bearachute at gmail dot com ¶

9 years ago

If you're having problems with binary and hex data:

I had a problem when trying to read information from a ttf, which is primarily hex data. A binary-safe file read automatically replaces byte values with their corresponding ASCII characters, so I thought that I could use the binary string when I needed readable ASCII strings, and bin2hex() when I needed hex strings.

However, this became a problem when I tried to pass those ASCII strings into other functions (namely gd functions). var_dump showed that a 5-character string contained 10 characters, but they weren't visible. A binary-to-"normal" string conversion function didn't seem to exist and I didn't want to have to convert every single character in hex using chr().

I used unpack with "c*" as the format flag to see what was going on, and found that every other character was null data (ordinal 0). To solve it, I just did

```
str_replace(chr(0), "", $string);
```

which did the trick.

This took forever to figure out so I hope this helps people reading from hex data!

up
down
6

siegfri3d at gmail dot com ¶

9 years ago

Use the previous example if you want to request the server for a special part of the content, IF and only if the server accepts the method.
If you want a simple example to ask the server for all the content, but only save a portion of it, do it this way:

```
<?php
$content=file_get_contents("http://www.google.com",FALSE,NULL,0,20);
echo $content;
?>
```

This will echo the 20 first bytes of the google.com source code.

up
down
2

andrew at 21cv dot co dot uk ¶

4 years ago

Negative offsets don't work as you might expect (like in http://php.net/substr for example)

So
<?php echo file_get_contents(__FILE__, false, null, -10) ?>
does the same as
<?php echo file_get_contents(__FILE__, false, null, 0) ?>

To get the last 10 characters of a file, you need to use
<?php echo file_get_contents (__FILE__, false, null, (filesize (__FILE__) - 10)) ?>
up
down
2
godwraith01 at yahoo dot com ¶
5 years ago
I experienced a problem in using hostnames instead straight IP with some server destinations.

If i use file_get_contents("www.jbossServer.example/app1",...)
i will get an 'Invalid hostname' from the server i'm calling.

This is because file_get_contents probably will rewrite your request after getting the IP, obtaining the same thing as :
file_get_contents("xxx.yyy.www.zzz/app1",...)

And you know that many servers will deny you access if you go through IP addressing in the request.

With cURL this problem doesn't exists. It resolves the hostname leaving the request as you set it, so the server is not rude in response.
up
down
-2
werdnanoslen at gmail dot com ¶
4 years ago
This is an easy way to trigger scripts by listening for POSTs. I simply point a service's webhook url to the script, which file_get_contents("php://input"), cast to an array, and then simplexml_load_string() to parse it and use one of the keys' data as the parameter for my script.
up
down
-1
Anonymous ¶
5 years ago
The offset is 0 based. Setting it to 1 will skip the first character of the stream.
up
down
0
3n1gm4 [at] gmail [dot] com ¶
8 years ago
This is a nice and simple substitute to get_file_contents() using curl, it returns FALSE if $contents is empty.

```
<?php
function curl_get_file_contents($URL)
    {
        $c = curl_init();
        curl_setopt($c, CURLOPT_RETURNTRANSFER, 1);
```

```
        curl_setopt($c, CURLOPT_URL, $URL);
        $contents = curl_exec($c);
        curl_close($c);

        if ($contents) return $contents;
            else return FALSE;
    }
?>
```

Hope this help, if there is something wrong or something you don't understand let me know :)

up
down
-2

luby dot pl at gmail dot com ¶

2 years ago

The funniest thing there is that seeking on non local files may, or may not work. This is unpredictable, and thus should throw rather than doing some magical stuff.
Also trying to read non local file which doesn't exists results in FALSE returned and no single warning emitted.

up
down
1

richard dot quadling at bandvulc dot co dot uk ¶

11 years ago

If, like me, you are on a Microsoft network with ISA server and require NTLM authentication, certain applications will not get out of the network. SETI@Home Classic and PHP are just 2 of them.

The workaround is fairly simple.

First you need to use an NTLM Authentication Proxy Server. There is one written in Python and is available from http://apserver.sourceforge.net/. You will need Python from http://www.python.org/.

Both sites include excellent documentation.

Python works a bit like PHP. Human readable code is handled without having to produce a compiled version. You DO have the opportunity of compiling the code (from a .py file to a .pyc file).

Once compiled, I installed this as a service (instsrv and srvany - parts of the Windows Resource Kit), so when the server is turned on (not logged in), the Python based NTLM Authentication Proxy Server is running.

Then, and here is the bit I'm really interested in, you need to tell PHP you intend to route http/ftp requests through the NTLM APS.

To do this, you use contexts.

Here is an example.

```
<?php

// Define a context for HTTP.
$aContext = array(
    'http' => array(
        'proxy' => 'tcp://127.0.0.1:8080', // This needs to be the server and the port of the NTLM
```

```
Authentication Proxy Server.
        'request_fulluri' => True,
        ),
    );
$cxContext = stream_context_create($aContext);

// Now all file stream functions can use this context.

$sFile = file_get_contents("http://www.php.net", False, $cxContext);

echo $sFile;
?>

Hopefully this helps SOMEONE!!!
```

up
down
-1

Greg Ambrose (greg at catalina-it dot com dot au) ¶

9 years ago

```
[Editors note: As of PHP 5.2.1 you can specify `timeout` context option and pass the context to
file_get_contents()]

The only way I could get get_file_contents() to wait for a very slow http request was to set the
socket timeout as follows.

ini_set('default_socket_timeout',    120);
$a = file_get_contents("http://abcxyz.com");

Other times like execution time and input time had no effect.
```

up
down
-3

rutger at webjin dot nl ¶

6 years ago

```
Sometimes you might get an error opening an http URL.
even though you have set "allow_url_fopen = On" in php.ini

For me the the solution was to also set "user_agent" to something.
```

up
down
-5

fibrefox at dynamicfiles dot de ¶

5 years ago

```
If you want to insert tracking-scripts into your shopping-system, some scripts doesn't support
intelligent detection of HTTPS, so i made a script i put on the server that rewrites 'http' to
'https' in the script, assuming everything has to be UTF-8 encoded (as a fallback it makes a
redirect).

It is important that the HTTPS-source DOES exist!

<?php

function file_get_contents_utf8($fn) {
    $opts = array(
```

```
        'http' => array(
            'method'=>"GET",
            'header'=>"Content-Type: text/html; charset=utf-8"
        )
    );

    $context = stream_context_create($opts);
    $result = @file_get_contents($fn,false,$context);
    return $result;
}
header("Content-Type: text/html; charset=utf-8");
$tPath = "URL YOU WANT TO MODIFY";
$result = file_get_contents_utf8("http://".$tPath);

if( $result == false){
    header("Location: https://".$tPath); // fallback
    exit();
}
else{
    echo mb_ereg_replace("http","https",$result);
}
?>
```

up
down
-3

aidan at php dot net ¶

11 years ago

This functionality is now implemented in the PEAR package PHP_Compat.

More information about using this function without upgrading your version of PHP can be found on the below link:

http://pear.php.net/package/PHP_Compat

up
down
-6

php [spat] hm2k.org ¶

8 years ago

I decided to make a similar function to this, called file_post_contents, it uses POST instead of GET to call, kinda handy...

```
<?php
function file_post_contents($url,$headers=false) {
    $url = parse_url($url);

    if (!isset($url['port'])) {
      if ($url['scheme'] == 'http') { $url['port']=80; }
      elseif ($url['scheme'] == 'https') { $url['port']=443; }
    }
    $url['query']=isset($url['query'])?$url['query']:'';

    $url['protocol']=$url['scheme'].'://';
    $eol="\r\n";
```

```
    $headers =  "POST ".$url['protocol'].$url['host'].$url['path']." HTTP/1.0".$eol.
                "Host: ".$url['host'].$eol.
                "Referer: ".$url['protocol'].$url['host'].$url['path'].$eol.
                "Content-Type: application/x-www-form-urlencoded".$eol.
                "Content-Length: ".strlen($url['query']).$eol.
                $eol.$url['query'];
    $fp = fsockopen($url['host'], $url['port'], $errno, $errstr, 30);
    if($fp) {
      fputs($fp, $headers);
      $result = '';
      while(!feof($fp)) { $result .= fgets($fp, 128); }
      fclose($fp);
      if (!$headers) {
        //removes headers
        $pattern="/^.*\r\n\r\n/s";
        $result=preg_replace($pattern,'',$result);
      }
      return $result;
    }
}
?>
```

up
down
-10
vlad dot wing at gmail dot com ¶
6 years ago

If you want to check if the function returned error, in case of a HTTP request an, it's not sufficient to test it against false. It may happen the return for that HTTP request was empty. In this case it's better to check if the return value is a bool.

```
<?php
$result=file_get_contents("http://www.example.com");
if ($result === false)
{
    // treat error
} else {
    // handle good case
}
?>
```

[EDIT BY thiago: Has enhacements from an anonymous user]

up
down
-12
pperegrina ¶
4 years ago

For those having this problem when trying to get_file_contents(url):

Warning: file_get_contents(url): failed to open stream: HTTP request failed! in xx on line yy

If you are behind a SonicWall firewall, read this:
https://bugs.php.net/bug.php?id=40197
(this little line: uncheck a box in the internal settings of the firewall labled "Enforce Host Tag Search with for CFS")

Apparently by default SonicWall blocks any HTTP request without a "Host:" header, which is the case in the PHP get_file_contents(url) implementation.

This is why, if you try to get the same URL from the same machine with cURL our wget, it works.

I hope this will be useful to someone, it took me hours to find out :)

up
down
-12

contact at webapp dot fr ¶
5 years ago

When using a URI with a login / password (HTTP or FTP, for an example), you may need to urlencode the password if it contains special characters.
Do not urlencode the whole URI, just the password.

Don't do :
```
urlencode('ftp://login:mdp%?special@host/dir/file')
```

Do :
```
'ftp://login:' . urlencode('mdp%?special') . '@host/dir/file';
```

Might seem obvious, but is worth noting.

up
down
-14

dustin at dumontproject dot com-spam sux ¶
4 years ago

For those who use file_get_contents for JSON or other RESTful services - like my architecture did for a big site - this will probably help a lot.

We struggled with having the site using get urls that would go through our load balancer instead of hitting the local server.

What we did was load this function through a local url and set the Host: header for our virtualhost entries on the site we wanted to laod.

This code solved our issue:
```
<?php

//set the header context stream for virtualhost lookup
$context = stream_context_create(array('http' => array('header' => 'Host:
www.VIRTUALHOSTDOMAIN.com')));

//use a localhost url or alternatively 127.0.0.1 ip
$url = 'http://localhost/rest-service/?get-user&id=######';

//fetch the data through webserver using the Host http header we set
$data = json_decode(file_get_contents($url, 0, $context));

//verify you have your data
var_dump($data);

?>
```

up
down
-14
ken at wetken dot net ¶
7 years ago
On Centos 5, and maybe other Red Hat based systems, any attempt to use file_get_contents to access a URL on an http port other than 80 (e.g. "http://www.example.com:8040/page") may fail with a permissions violation (error 13) unless the box you are running php on has its seLinux set to 'permissive' not 'enforcing' . Otherwise the request doesn't even get out of the box, i.e. the permissions violation is generated locally by seLinux.
up
down
-17
corey at effim dot com ¶
7 years ago
In my dev environment with a relatively low-speed drive (standard SATA 7200RPM) reading a 25MB zip file in 10 times...

```
<?php

$data = `cat /tmp/test.zip`;
// 1.05 seconds

$fh = fopen('/tmp/test.zip', 'r');
$data = fread($fh, filesize('/tmp/test.zip'));
fclose($fh);
// 1.31 seconds

$data = file_get_contents('/tmp/test.zip');
// 1.33 seconds

?>
```

However, on a 21k text file running 100 iterations...

```
<?php

$data = `cat /tmp/test.txt`;
// 1.98 seconds

$fh = fopen('/tmp/test.txt', 'r');
$data = fread($fh, filesize('/tmp/test.txt'));
fclose($fh);
// 0.00082 seconds

$data = file_get_contents('/tmp/test.txt');
// 0.0069 seconds

?>
```

Despite the comment about file_get_contents being faster do to memory mapping, file_get_contents is slowest in both of the above examples. If you need the best performance out of your production box, you might want to throw together a script to check out which method is fastest for what size files on

that particular machine, then optimize your code to check the file size and use the appropriate function for it.

up
down
-10

srgold at hotmail dot com ¶

2 years ago

Be sure to remove newlines from variables when using file_get_contents, or you'll receive the following error:

file_get_contents(): failed to open stream: HTTP request failed! HTTP/1.1 404 Not Found

up
down
-20

fcicqbbs at gmail dot com ¶

10 years ago

the bug #36857 was fixed.
http://bugs.php.net/36857

Now you may use this code,to fetch the partial content like this:

```
<?php
$context=array('http' => array ('header'=> 'Range: bytes=1024-', ),);
$xcontext = stream_context_create($context);
$str=file_get_contents("http://www.fcicq.net/wp/",FALSE,$xcontext);
?>
```

that's all.

up
down
-48

leomac at inbox dot ru ¶

4 years ago

Reading all script input is simple task with file_get_contents, but it depends on what SAPI is being used.

Only in Apache, not in CLI:

```
<?php
  $input = file_get_contents("php://input");
?>
```

Only in CLI, not in Apache:

```
<?php
  $input = file_get_contents("php://stdin");
?>
```

In Apache php://stdin will be empty, in CLI php://input will be empyt instead with no error indication.

add a note

- Filesystem Functions
  - basename
  - chgrp
  - chmod
  - chown
  - clearstatcache

- copy
- delete
- dirname
- disk_free_space
- disk_total_space
- diskfreespace
- fclose
- feof
- fflush
- fgetc
- fgetcsv
- fgets
- fgetss
- file_exists
- file_get_contents
- file_put_contents
- file
- fileatime
- filectime
- filegroup
- fileinode
- filemtime
- fileowner
- fileperms
- filesize
- filetype
- flock
- fnmatch
- fopen
- fpassthru
- fputcsv
- fputs
- fread
- fscanf
- fseek
- fstat
- ftell
- ftruncate
- fwrite
- glob
- is_dir
- is_executable
- is_file
- is_link
- is_readable
- is_uploaded_file
- is_writable
- is_writeable
- lchgrp
- lchown
- link
- linkinfo
- lstat
- mkdir

- move_uploaded_file
- parse_ini_file
- parse_ini_string
- pathinfo
- pclose
- popen
- readfile
- readlink
- realpath_cache_get
- realpath_cache_size
- realpath
- rename
- rewind
- rmdir
- set_file_buffer
- stat
- symlink
- tempnam
- tmpfile
- touch
- umask
- unlink