php

- Downloads
- Documentation
- Get Involved
- Help

Search

PHP 7.2.0 Release Candidate 4 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

g h
: Goto homepage

get_browser

(PHP 4, PHP 5, PHP 7)

get_browser — Tells what the user's browser is capable of

Description ¶

mixed **get_browser** ([string `$user_agent` [, bool `$return_array` = false]])

Attempts to determine the capabilities of the user's browser, by looking up the browser's information in the *browscap.ini* file.

Parameters ¶

`user_agent`

The User Agent to be analyzed. By default, the value of HTTP User-Agent header is used; however, you can alter this (i.e., look up another browser's info) by passing this parameter.

You can bypass this parameter with a **`NULL`** value.

`return_array`

If set to **`TRUE`**, this function will return an array instead of an object.

Return Values ¶

The information is returned in an object or an array which will contain various data elements representing, for instance, the browser's major and minor version numbers and ID string; **`TRUE`**/**`FALSE`** values for features such as frames, JavaScript, and cookies; and so forth.

The *cookies* value simply means that the browser itself is capable of accepting cookies and does not mean the user has enabled the browser to accept cookies or not. The only way to test if cookies are accepted is to set one with setcookie(), reload, and check for the value.

Examples ¶

Example #1 Listing all information about the users browser

```php
<?php
echo $_SERVER['HTTP_USER_AGENT'] . "\n\n";

$browser = get_browser(null, true);
print_r($browser);
?>
```

The above example will output something similar to:

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7) Gecko/20040803 Firefox/0.9.3

Array
(
    [browser_name_regex] => ^mozilla/5\.0 (windows; .; windows nt 5\.1; .*rv:.*) gecko/.* firefox/0\.9.*$
    [browser_name_pattern] => Mozilla/5.0 (Windows; ?; Windows NT 5.1; *rv:*) Gecko/* Firefox/0.9*
    [parent] => Firefox 0.9
    [platform] => WinXP
    [browser] => Firefox
    [version] => 0.9
    [majorver] => 0
    [minorver] => 9
    [cssversion] => 2
    [frames] => 1
    [iframes] => 1
    [tables] => 1
    [cookies] => 1
    [backgroundsounds] =>
    [vbscript] =>
    [javascript] => 1
    [javaapplets] => 1
    [activexcontrols] =>
    [cdf] =>
    [aol] =>
    [beta] => 1
    [win16] =>
    [crawler] =>
    [stripper] =>
    [wap] =>
    [netclr] =>
)
```

Notes ¶

> **Note**:
>
> In order for this to work, your browscap configuration setting in *php.ini* must point to the correct location of the *browscap.ini* file on your system.
>
> *browscap.ini* is not bundled with PHP, but you may find an up-to-date » php_browscap.ini file here.
>
> While *browscap.ini* contains information on many browsers, it relies on user updates to keep the database current. The format of the file is fairly self-explanatory.

add a note

User Contributed Notes 37 notes

up
down
86
Anonymous ¶
1 year ago

This function is too slow for todays needs.

If you need browser / device / operating system detection, please try one of listed packages here: https://github.com/ThaDafinser/UserAgentParser

up
down
31

ruudrp at live dot nl ¶
6 years ago

To my surprise I found that none of the get_browser alternatives output the correct name / version combination that I was looking for using Opera or Chrome. They either give the wrong name eg Safari when in fact it should be Chrome and if the ua string includes a version number as with the latest versions of Chrome and Opera the wrong number is reported. So I took bits and pieces from the various examples and combined them and added a check for version.

```
<?php
function getBrowser()
{
    $u_agent = $_SERVER['HTTP_USER_AGENT'];
    $bname = 'Unknown';
    $platform = 'Unknown';
    $version= "";

    //First get the platform?
    if (preg_match('/linux/i', $u_agent)) {
        $platform = 'linux';
    }
    elseif (preg_match('/macintosh|mac os x/i', $u_agent)) {
        $platform = 'mac';
    }
    elseif (preg_match('/windows|win32/i', $u_agent)) {
        $platform = 'windows';
    }

    // Next get the name of the useragent yes seperately and for good reason
    if(preg_match('/MSIE/i',$u_agent) && !preg_match('/Opera/i',$u_agent))
    {
        $bname = 'Internet Explorer';
        $ub = "MSIE";
    }
    elseif(preg_match('/Firefox/i',$u_agent))
    {
        $bname = 'Mozilla Firefox';
        $ub = "Firefox";
    }
    elseif(preg_match('/Chrome/i',$u_agent))
    {
        $bname = 'Google Chrome';
        $ub = "Chrome";
    }
    elseif(preg_match('/Safari/i',$u_agent))
    {
        $bname = 'Apple Safari';
        $ub = "Safari";
    }
    elseif(preg_match('/Opera/i',$u_agent))
    {
        $bname = 'Opera';
```

```
        $ub = "Opera";
    }
    elseif(preg_match('/Netscape/i',$u_agent))
    {
        $bname = 'Netscape';
        $ub = "Netscape";
    }

    // finally get the correct version number
    $known = array('Version', $ub, 'other');
    $pattern = '#(?<browser>' . join('|', $known) .
    ')[/ ]+(?<version>[0-9.|a-zA-Z.]*)#';
    if (!preg_match_all($pattern, $u_agent, $matches)) {
        // we have no matching number just continue
    }

    // see how many we have
    $i = count($matches['browser']);
    if ($i != 1) {
        //we will have two since we are not using 'other' argument yet
        //see if version is before or after the name
        if (strripos($u_agent,"Version") < strripos($u_agent,$ub)){
            $version= $matches['version'][0];
        }
        else {
            $version= $matches['version'][1];
        }
    }
    else {
        $version= $matches['version'][0];
    }

    // check if we have a number
    if ($version==null || $version=="") {$version="?";}

    return array(
        'userAgent' => $u_agent,
        'name'      => $bname,
        'version'   => $version,
        'platform'  => $platform,
        'pattern'    => $pattern
    );
}

// now try it
$ua=getBrowser();
$yourbrowser= "Your browser: " . $ua['name'] . " " . $ua['version'] . " on " .$ua['platform'] . " reports: <br >" . $ua['userAgent'];
print_r($yourbrowser);
?>
```

up
down
11
Francesco R ¶
1 year ago

If you ONLY need a very fast and simple function to detect the browser name (update to May 2016):

```
<?php

function get_browser_name($user_agent)
{
    if (strpos($user_agent, 'Opera') || strpos($user_agent, 'OPR/')) return 'Opera';
    elseif (strpos($user_agent, 'Edge')) return 'Edge';
    elseif (strpos($user_agent, 'Chrome')) return 'Chrome';
    elseif (strpos($user_agent, 'Safari')) return 'Safari';
    elseif (strpos($user_agent, 'Firefox')) return 'Firefox';
    elseif (strpos($user_agent, 'MSIE') || strpos($user_agent, 'Trident/7')) return 'Internet Explorer';

    return 'Other';
}

// Usage:

echo get_browser_name($_SERVER['HTTP_USER_AGENT']);

?>
```

This function also resolves the trouble with Edge (that contains in the user agent the string "Safari" and "Chrome"), with Chrome (contains the string "Safari") and IE11 (that do not contains 'MSIE' like all other IE versions).

Note that "strpos" is the fastest function to check a string (far better than "preg_match") and Opera + Edge + Chrome + Safari + Firefox + Internet Explorer are the most used browsers today (over 97%).

up
down
4

tim at digicol dot de ¶
3 years ago

Be aware that loading php_browscap.ini via the browscap php.ini setting may consume a non-trivial amount of memory. Current versions are several MB in size (even the “lite” one) and can eat tens of MB of RAM in each PHP process. This happens even if you never call get_browser() since php_browscap.ini is loaded when PHP is starting up.

Make sure to leave the browscap php.ini setting empty if you don’t use get_browser() – maybe you only call it only from PHP Web pages, but not from PHP CLI code.

I’d recommend comparing your processes’ memory consumption with and without php_browscap.ini being loaded. If necesseray, consider creating your own stripped-down copy of php_browscap.ini with just the browsers that are important to you.

up
down
2

Anonymous ¶
6 years ago

For those of you using this function to target MSIE, a better idea maybe to use MSIE specific conditional comments. More info: <http://msdn.microsoft.com/en-us/library/ms537512%28VS.85%29.aspx>.

For example to indicate your disregard for users of MSIE 6 or earlier:

```
<!--[if lt IE 7]>It appears that you are using a <em>very</em> old version of MS Internet Explorer (MSIE). If you seriously want to continue to use MSIE, at least <a href="http://www.microsoft.com/windows/internet-explorer/">upgrade</a>.<![endif]-->
```

up
down
1

max at phpexpert dot de ¶
13 years ago
Be aware of the fact that this function shows what a specific browser might be able to show, but NOT what the user has turned on/off.

So maybe this function tells you that the browser is abel to to javascript even when javascript is turned off by the user.
up
down
0
The Digital Orchard ¶
8 months ago
Good news! The latest version of PHP has a performance fix for this function. It's reportedly now 100x faster. See the ChangeLog for specifics.
up
down
0
zed ¶
1 year ago
To complete Francesco R, I added the version of the navigator :

```
function getNavigateur($user_agent)
    {

        if(empty($user_agent)) {
            return array('nav' => 'NC', 'name' => 'NC', 'version' => 'NC');
        }

        $content_nav['name'] = 'Unknown';

        if (strpos($user_agent, 'Opera') || strpos($user_agent, 'OPR/')) {

            $content_nav['name'] = 'Opera';

            if (strpos($user_agent, 'OPR/')) {
                $content_nav['reel_name'] = 'OPR/';
            } else {
                $content_nav['reel_name'] = 'Opera';
            }

        }
        elseif (strpos($user_agent, 'Edge')) {
            $content_nav['name'] = $content_nav['reel_name'] = 'Edge';
        }
        elseif (strpos($user_agent, 'Chrome')) $content_nav['name'] = $content_nav['reel_name'] = 'Chrome';
        elseif (strpos($user_agent, 'Safari')) $content_nav['name'] = $content_nav['reel_name'] = 'Safari';
        elseif (strpos($user_agent, 'Firefox')) $content_nav['name'] = $content_nav['reel_name'] = 'Firefox';
        elseif (strpos($user_agent, 'MSIE') || strpos($user_agent, 'Trident/7') || strpos($user_agent, 'Trident/7.0; rv:')) {
            $content_nav['name'] = 'Internet Explorer';

            if (strpos($user_agent, 'Trident/7.0; rv:')) {
                $content_nav['reel_name'] = 'Trident/7.0; rv:';
            } elseif (strpos($user_agent, 'Trident/7')) {
                $content_nav['reel_name'] = 'Trident/7';
            } else {
                $content_nav['reel_name'] = 'Opera';
            }
```

```
        }

        $pattern = '#' . $content_nav['reel_name'] . '\/*([0-9\.]*)#';

        $matches = array();

        if(preg_match($pattern, $user_agent, $matches)) {

            $content_nav['version'] = $matches[1];
            return $content_nav;

        }

        return array('name' => $content_nav['name'], 'version' => 'Inconnu');
    }
```

up
down
0
p2 at eduardoruiz dot es ¶
1 year ago

BE CAREFUL WITH THIS FUNCTION!!
This function uses a lot of CPU and RAM on the whole server resources.
Perhaps if you use this function a few times then no problem, but NOT if you use at any page request, or once per session.

Also, this function doesn't work correctly and may returns wrong values, wildcards or empty, so it's not very useful for web statistics.

The best way is to use preg_match for detect browser/platform.

up
down
-1
shashank ¶
2 years ago

As ruudrp had given the code http://php.net/manual/en/function.get-browser.php#101125, I have added code for Internet Explorer 11

```
<?php
function getBrowser()
{
    $u_agent = $_SERVER['HTTP_USER_AGENT'];
    $bname = 'Unknown';
    $platform = 'Unknown';
    $version= "";

    //First get the platform?
    if (preg_match('/linux/i', $u_agent)) {
        $platform = 'linux';
    }
    elseif (preg_match('/macintosh|mac os x/i', $u_agent)) {
        $platform = 'mac';
    }
    elseif (preg_match('/windows|win32/i', $u_agent)) {
        $platform = 'windows';
    }

    // Next get the name of the useragent yes seperately and for good reason
    if(preg_match('/MSIE/i',$u_agent) && !preg_match('/Opera/i',$u_agent))
```

```
    {
        $bname = 'Internet Explorer';
        $ub = "MSIE";
    }
    elseif(preg_match('/Trident/i',$u_agent))
    { // this condition is for IE11
        $bname = 'Internet Explorer';
        $ub = "rv";
    }
    elseif(preg_match('/Firefox/i',$u_agent))
    {
        $bname = 'Mozilla Firefox';
        $ub = "Firefox";
    }
    elseif(preg_match('/Chrome/i',$u_agent))
    {
        $bname = 'Google Chrome';
        $ub = "Chrome";
    }
    elseif(preg_match('/Safari/i',$u_agent))
    {
        $bname = 'Apple Safari';
        $ub = "Safari";
    }
    elseif(preg_match('/Opera/i',$u_agent))
    {
        $bname = 'Opera';
        $ub = "Opera";
    }
    elseif(preg_match('/Netscape/i',$u_agent))
    {
        $bname = 'Netscape';
        $ub = "Netscape";
    }

    // finally get the correct version number
    // Added "|:"
    $known = array('Version', $ub, 'other');
    $pattern = '#(?<browser>' . join('|', $known) .
     ')[/|: ]+(?<version>[0-9.|a-zA-Z.]*)#';
    if (!preg_match_all($pattern, $u_agent, $matches)) {
        // we have no matching number just continue
    }

    // see how many we have
    $i = count($matches['browser']);
    if ($i != 1) {
        //we will have two since we are not using 'other' argument yet
        //see if version is before or after the name
        if (strripos($u_agent,"Version") < strripos($u_agent,$ub)){
            $version= $matches['version'][0];
        }
        else {
            $version= $matches['version'][1];
        }
```

```
    }
    else {
        $version= $matches['version'][0];
    }

    // check if we have a number
    if ($version==null || $version=="") {$version="?";}

    return array(
        'userAgent' => $u_agent,
        'name'      => $bname,
        'version'   => $version,
        'platform'  => $platform,
        'pattern'    => $pattern
    );
}

// now try it
$ua=getBrowser();
$yourbrowser= "Your browser: " . $ua['name'] . " " . $ua['version'] . " on " .$ua['platform'] . " reports: <br >" . $ua['userAgent'];
print_r($yourbrowser);
?>
```

up
down
-1
Steffen ¶
9 years ago
Keep in mind that get_browser(); really slows down your application. It takes about 22 ms to execute on an idle server with Ubuntu Linux, Apache 2, PHP 5.1.3.

up
down
-2
bishop ¶
14 years ago
PHP is sensitive to characters outside the range [A-Za-z0-9_] as values in .ini files. For example

```
browser=Opera (Bork Version)
```

causes PHP to complain, as it doesn't like the parentheses.

If you place quotation marks around the values for all keys in the browscap.ini file, you'll save yourself parsing problems. Do this in eg vi with %s/=\(.*\)/="\1"/g

You could of course use PHP itself to fixup the file. Exercise left to the reader.

up
down
-1
mike at mike-griffiths dot co dot uk ¶
10 years ago
You should not rely on just this for cross-browser compatibility issues. Good practice would be to include HTML if-statements for IE stylesheets as well as dynamically checking the browser type.

up
down
-6
ibi at tlr dot de ¶
3 years ago

```
@ruudrp:
you might want to distinguish the OS even further:
(at least for windows that is)

    elseif (preg_match('/windows|win32/i', $u_agent)) {
        $platform = 'Windows';
        if (preg_match('/NT 6.2/i', $u_agent)) { $platform .= ' 8'; }
            elseif (preg_match('/NT 6.3/i', $u_agent)) { $platform .= ' 8.1'; }
            elseif (preg_match('/NT 6.1/i', $u_agent)) { $platform .= ' 7'; }
            elseif (preg_match('/NT 6.0/i', $u_agent)) { $platform .= ' Vista'; }
            elseif (preg_match('/NT 5.1/i', $u_agent)) { $platform .= ' XP'; }
            elseif (preg_match('/NT 5.0/i', $u_agent)) { $platform .= ' 2000'; }
        if (preg_match('/WOW64/i', $u_agent) || preg_match('/x64/i', $u_agent)) { $platform .= ' (x64)'; }
    }
```

up
down
-7
robert at broofa dot com ¶
8 years ago

If you're just finding this API, note that you may want to use a lighter-weight browser detection script. get_browser() requires the "browscap.ini" file, which is 300KB+. Loading and processing this file will likely impact script performance. Although it surely provides excellent detection results, in most cases a much simpler method can be just as effective. This is why so many previous commenters have provided alternate implementations.

Here's the solution I ended up using, which I've tested on the agents listed at http://whatsmyuseragent.com/CommonUserAgents.asp It has the advantage of being compact and reasonably easy to extend (just add entries to the $known array defined at the top). It should be fairly performant as well, since it doesn't do any iteratoin or recursion.

```
<?php
function browser_info($agent=null) {
  // Declare known browsers to look for
  $known = array('msie', 'firefox', 'safari', 'webkit', 'opera', 'netscape',
    'konqueror', 'gecko');

  // Clean up agent and build regex that matches phrases for known browsers
  // (e.g. "Firefox/2.0" or "MSIE 6.0" (This only matches the major and minor
  // version numbers.  E.g. "2.0.0.6" is parsed as simply "2.0"
  $agent = strtolower($agent ? $agent : $_SERVER['HTTP_USER_AGENT']);
  $pattern = '#(?<browser>' . join('|', $known) .
    ')[/ ]+(?<version>[0-9]+(?:\.[0-9]+)?)#';

  // Find all phrases (or return empty array if none found)
  if (!preg_match_all($pattern, $agent, $matches)) return array();

  // Since some UAs have more than one phrase (e.g Firefox has a Gecko phrase,
  // Opera 7,8 have a MSIE phrase), use the last one found (the right-most one
  // in the UA).  That's usually the most correct.
  $i = count($matches['browser'])-1;
  return array($matches['browser'][$i] => $matches['version'][$i]);
}
?>
```

```
This returns an array with the detected browser as the key, and the version as
the value, and also sets 'browser' and 'version' keys.  For example on Firefox
3.5:
<?php
$ua = browser_info();
print_r($ua);
/* Yields ...
Array
(
    [firefox] => 3.5
    [browser] => firefox
    [version] => 3.5
)
*/

// Various browser tests you can do with the returned array ...
if ($ua['firefox']) ... // true
if ($ua['firefox'] > 3) ... // true
if ($ua['firefox'] > 4) ... // false
if ($ua['browser'] == 'firefox') ... // true
if ($ua['version'] > 3.5) ... // true
if ($ua['msie']) ... // false ('msie' key not defined)
if ($ua['opera'] > 3) ... // false ('opera' key not defined)
if ($ua['safari'] < 3) ... // false also ('safari' key not defined)
?>
```

up
down
-3

Becheru Petru-Ioan ¶
7 years ago

IE has a nasty bug called the Peekaboo bug that affected my website. I found that printing a '_' before the html tag of the webpage gets rid of this nasty bug. Using code inspirated by comment bellow here is the code that detects if a visitor is using internet explorer:

```
<?php
$known = array('msie', 'firefox', 'safari', 'webkit', 'opera', 'netscape', 'konqueror', 'gecko');
preg_match_all( '#(?<browser>' . join('|', $known) .
    ')[/ ]+(?<version>[0-9]+(?:\.[0-9]+)?)#', strtolower( $_SERVER[ 'HTTP_USER_AGENT ' ]), $browser );
if($browser['browser'][0]=='msie') print('_');
?>
```

up
down
-6

Brinley Ang ¶
7 years ago

This is the latest format of the array returned from this call

```
Array
{
    [browser_name_regex] => ^mozilla/5\.0 (x11; .*; .*linux.*; .*; rv:1\.9\..*) gecko/.* firefox/3\.5.*$
    [browser_name_pattern] => Mozilla/5.0 (X11; *; *Linux*; *; rv:1.9.*) Gecko/* Firefox/3.5*
    [parent] => Firefox 3.5
    [platform] => Linux
    [browser] => Firefox
    [version] => 3.5
    [majorver] => 3
```

```
    [minorver] => 5
    [frames] => 1
    [iframes] => 1
    [tables] => 1
    [cookies] => 1
    [javaapplets] => 1
    [javascript] => 1
    [cssversion] => 3
    [supportscss] => 1
    [alpha] =>
    [beta] =>
    [win16] =>
    [win32] =>
    [win64] =>
    [backgroundsounds] =>
    [cdf] =>
    [vbscript] =>
    [activexcontrols] =>
    [isbanned] =>
    [ismobiledevice] =>
    [issyndicationreader] =>
    [crawler] =>
    [aol] =>
    [aolversion] => 0
}
```

up
down
-8
cbonfrate at gmail dot com ¶
3 years ago

You could add this to the ruudrp's code, to recognize MSIE 11

```
if(preg_match('/Trident/i', $u_agent) && !preg_match('/Opera/i',$u_agent))
    {
        $bname = 'Internet Explorer';
        $ub = "Trident";
    }
```

up
down
-6
sam ¶
10 years ago

I thought this function might be useful to those without access to the php.ini file (such as those on a shared hosting system):

```
<?php
function php_get_browser($agent = NULL){
$agent=$agent?$agent:$_SERVER['HTTP_USER_AGENT'];
$yu=array();
$q_s=array("#\.#","#\*#","#\?#");
$q_r=array("\.",".*",".?");
$brows=parse_ini_file("php_browscap.ini",true);
foreach($brows as $k=>$t){
  if(fnmatch($k,$agent)){
  $yu['browser_name_pattern']=$k;
  $pat=preg_replace($q_s,$q_r,$k);
```

```
   $yu['browser_name_regex']=strtolower("^$pat$");
     foreach($brows as $g=>$r){
       if($t['Parent']==$g){
         foreach($brows as $a=>$b){
           if($r['Parent']==$a){
             $yu=array_merge($yu,$b,$r,$t);
             foreach($yu as $d=>$z){
               $l=strtolower($d);
               $hu[$l]=$z;
             }
           }
         }
       }
     }
     break;
   }
 }
 return $hu;
 }
 ?>

 define the location of php_browscap.ini wherever you want
 always returns an array, same functionality as get_browser(NULL,true)
 Hope someone finds it useful!
```

up
down
-5
Steve Perkins ¶
8 years ago

```
This is a simple class to detect the client browser and version using regular expressions.

<?PHP
class Browser
{
    private $props    = array("Version" => "0.0.0",
                              "Name" => "unknown",
                              "Agent" => "unknown") ;

    public function __Construct()
    {
        $browsers = array("firefox", "msie", "opera", "chrome", "safari",
                          "mozilla", "seamonkey",    "konqueror", "netscape",
                          "gecko", "navigator", "mosaic", "lynx", "amaya",
                          "omniweb", "avant", "camino", "flock", "aol");

        $this->Agent = strtolower($_SERVER['HTTP_USER_AGENT']);
        foreach($browsers as $browser)
        {
            if (preg_match("#($browser)[/ ]?([0-9.]*)#", $this->Agent, $match))
            {
                $this->Name = $match[1] ;
                $this->Version = $match[2] ;
                break ;
            }
        }
```

```
    }

    public function __Get($name)
    {
        if (!array_key_exists($name, $this->props))
        {
            die "No such property or function $name)" ;
        }
        return $this->props[$name] ;
    }

    public function __Set($name, $val)
    {
        if (!array_key_exists($name, $this->props))
        {
            SimpleError("No such property or function.", "Failed to set $name", $this->props) ;
            die ;
        }
        $this->props[$name] = $val ;
    }

}

?>

example code
<?PHP
$browser = new Browser ;
echo "$Browser->Name $Browser->Version" ;
?>
result when client using Firefox 3.0.11
firefox 3.0.11

result when client using unknown browser
unknown 0.0.0

etc etc
```

up
down
-9
jesdisciple [at] gmail -dawt- com ¶
9 years ago

```
Use this to ensure that the costly call in its standard form never needs to be repeated:
<?php
function getBrowser(){
    static $browser;//No accident can arise from depending on an unset variable.
    if(!isset($browser)){
        $browser = get_browser($_SERVER['HTTP_USER_AGENT']);
    }
    return $browser;
}
?>
```

up
down
-1

jeremie dot legrand at komori-chambon dot fr ¶
1 year ago
Be careful if you use the "Full" Browscap INI file in your php.ini config: I wondered why each Apache thread took 350MB RAM on my server until I changed the "Full" version by the "Lite" one (45MB to 0.7MB)

Now, each thread takes only 16MB...
So if it is enough for you, use the Lite version!
up
down
-2
haci at muratyaman dot co dot uk ¶
3 years ago
I've just lost a day because of this stupid browser_name_regex property:

e.g. "§^mozilla/5\.0 \(.*windows nt 6\.1.*wow64.*\) gecko/.* firefox/29\.0.*$§"

To replicate the issue and work around bloody '§' character, please below. Whose idea was it to use that character?!

```
<?php
$ua = $_SERVER['HTTP_USER_AGENT'];
echo '$_SERVER[\'HTTP_USER_AGENT\'] => "' . $ua . '"<br />';

$browser = get_browser($ua);
echo 'get_browser($ua) => <pre>';
var_dump($browser);
echo '</pre><br />';

$json = json_encode($browser, JSON_PRETTY_PRINT);
echo 'json_encode($browser) => <pre>';
echo $json;
echo '</pre><br />';

echo 'json_last_error() => "' . json_last_error() . '"<br />';

//echo 'unsetting $browser->browser_name_regex <br />';
//unset($browser->browser_name_regex);
//or
echo '$browser->browser_name_regex = utf8_encode($browser->browser_name_regex);<br />';
$browser->browser_name_regex = utf8_encode($browser->browser_name_regex);

$json = json_encode($browser, JSON_PRETTY_PRINT);
echo 'json_encode($browser) => <pre>';
echo $json;
echo '</pre><br />';

echo 'json_last_error() => "' . json_last_error() . '"<br />';
?>
```

up
down
-4
nick at category4 dot com ¶
16 years ago
Here's a quick way to test for a Netscape browser. IE and Konqueror and several others call themselves "Mozilla", but they always qualify it with the word "compatible."

```
$isns = stristr($HTTP_USER_AGENT, "Mozilla") && (!(stristr($HTTP_USER_AGENT, "compatible")));
```

up
down
-4
henda dot exe at gmail dot com ¶
7 years ago

Just a warning to anybody using the preg_match function to determine what browser the viewer is using. I found with google chrome the array contains the following entries;
(Windows; U; Windows NT 6.1; en-US) AppleWebKit/533.4 (KHTML, like Gecko) Chrome/5.0.375.126 Safari/533.4

As I was using preg_match('/Chrome/i',$u_agent)) and preg_match('/Safari/i',$u_agent)) i found my script was reporting browsers as safari when infact the browser in use was google chrome.

up
down
-6
meduza ¶
5 years ago

I stopped trying to detect the browser name and the browser's version, due to the non standard user-agents, is impossible to cover all the possible browsers, and cover only the most used ones is not good.
I only detect the platform, the engine and the version, that cover a lot of possible browsers and let me code based on that, for example, the engine gecko, covers firefox, seamonkey and all the gecko based browsers.

this is a example piece of code taken from my own, is maybe incomplete but as example is good, to detect the engines.

```
<?php
/****** browser engine detection **************/
//browsers
define("UNKNOWN", 0);
define("TRIDENT", 1);
define("GECKO", 2);
define("PRESTO", 3);
define("WEBKIT", 4);
define("VALIDATOR", 5);
define("ROBOTS", 6);

if(!isset($_SESSION["info"]['browser'])) {
    $_SESSION["info"]['browser']['engine'] = UNKNOWN;
    $_SESSION["info"]['browser']['version'] = UNKNOWN;
    $_SESSION["info"]['browser']['platform'] = 'Unknown';

    $navigator_user_agent = ' ' . strtolower($_SERVER['HTTP_USER_AGENT']);

    if (strpos($navigator_user_agent, 'linux')) :
        $_SESSION["info"]['browser']['platform'] = 'Linux';
    elseif (strpos($navigator_user_agent, 'mac')) :
        $_SESSION["info"]['browser']['platform'] = 'Mac';
    elseif (strpos($navigator_user_agent, 'win')) :
        $_SESSION["info"]['browser']['platform'] = 'Windows';
    endif;

    if (strpos($navigator_user_agent, "trident")) {
        $_SESSION["info"]['browser']['engine'] = TRIDENT;
        $_SESSION["info"]['browser']['version'] = floatval(substr($navigator_user_agent, strpos($navigator_user_agent, "trident/") + 8, 3));
    }
```

```
    elseif (strpos($navigator_user_agent, "webkit")) {
        $_SESSION["info"]['browser']['engine'] = WEBKIT;
        $_SESSION["info"]['browser']['version'] = floatval(substr($navigator_user_agent, strpos($navigator_user_agent, "webkit/") + 7, 8));
    }
        elseif (strpos($navigator_user_agent, "presto")) {
        $_SESSION["info"]['browser']['engine'] = PRESTO;
        $_SESSION["info"]['browser']['version'] = floatval(substr($navigator_user_agent, strpos($navigator_user_agent, "presto/") + 6, 7));
    }
    elseif (strpos($navigator_user_agent, "gecko")) {
        $_SESSION["info"]['browser']['engine'] = GECKO;
        $_SESSION["info"]['browser']['version'] = floatval(substr($navigator_user_agent, strpos($navigator_user_agent, "gecko/") + 6, 9));
    }
    elseif (strpos($navigator_user_agent, "robot"))
        $_SESSION["info"]['browser']['engine'] = ROBOTS;
    elseif (strpos($navigator_user_agent, "spider"))
        $_SESSION["info"]['browser']['engine'] = ROBOTS;
    elseif (strpos($navigator_user_agent, "bot"))
        $_SESSION["info"]['browser']['engine'] = ROBOTS;
    elseif (strpos($navigator_user_agent, "crawl"))
        $_SESSION["info"]['browser']['engine'] = ROBOTS;
    elseif (strpos($navigator_user_agent, "search"))
        $_SESSION["info"]['browser']['engine'] = ROBOTS;
    elseif (strpos($navigator_user_agent, "w3c_validator"))
        $_SESSION["info"]['browser']['engine'] = VALIDATOR;
    elseif (strpos($navigator_user_agent, "jigsaw"))
        $_SESSION["info"]['browser']['engine'] = VALIDATOR;

echo "<pre>\n\nEngine detected: ".$_SESSION["info"]['browser']['engine'];
switch($_SESSION["info"]['browser']['engine']) {
    case UNKNOWN: echo " (unknown)";
    break;
    case TRIDENT: echo " (trident)";
    break;
    case GECKO: echo " (gecko)";
    break;
    case PRESTO: echo " (presto)";
    break;
    case WEBKIT: echo " (Webkit)";
    break;
    case VALIDATOR: echo " (validator)";
    break;
    case ROBOTS: echo " (robot)";
}
echo "\nEngine version: ".$_SESSION["info"]['browser']['version'];
echo "\nPlataform detected: ".$_SESSION["info"]['browser']['platform'];
echo " \n\n</pre>";
?>

PD: you can add more engines and use an array, I just removed that and put the most popular engines as example, and used elseif instead, this code maybe have errors
because is just an example.
```

up
down
-5
Steve Perkins ¶
8 years ago

This is a simple class to detect the client browser and version using regular expressions.

```
<?PHP
class Browser extends BaseObjects_PropertyArray
{
    private $props    = array("Version" => "0.0.0",
                              "Name" => "unknown",
                              "Agent" => "unknown",
                              "AllowsHeaderRedirect" => true) ;

    public function __Construct()
    {
        $browsers = array("firefox", "msie", "opera", "chrome", "safari",
                          "mozilla", "seamonkey",    "konqueror", "netscape",
                          "gecko", "navigator", "mosaic", "lynx", "amaya",
                          "omniweb", "avant", "camino", "flock", "aol");

        $this->Agent = strtolower($_SERVER['HTTP_USER_AGENT']);
        foreach($browsers as $browser)
        {
            if (preg_match("#($browser)[/ ]?([0-9.]*)#", $this->Agent, $match))
            {
                $this->Name = $match[1] ;
                $this->Version = $match[2] ;
                break ;
            }
        }
        $this->AllowsHeaderRedirect = !($this->Name == "msie" && $this->Version < 7) ;
    }

    public function __Get($name)
    {
        if (!array_key_exists($name, $this->props))
        {
            die "No such property or function $name)" ;
        }
        return $this->props[$name] ;
    }

}

?>

example code
<?PHP
$browser = new Browser ;
echo "$Browser->Name $Browser->Version" ;
?>
result when client using Firefox 3.0.11
firefox 3.0.11

result when client using unknown browser
unknown 0.0.0

etc etc
```

up
down
-6
Anonymous ¶
3 years ago
Warning!
For All Users there is a new website about new
"Browser Capabilities Project"
browscap.org

look here : goo.gl/gOPxHp
this is a question about one of the problems have been fixed in the new version.

تحذير!
لكل المستخدمين انه هناك موقع جديد عن مشروع قدرات المتصفح
"Browser Capabilities Project"
browscap.org

انظر هنا : goo.gl/gOPxHp
هذا هو سؤال عن احد المشاكل التى تم اصلاحُها فى الاصدار الجديد

up
down
-5
triad at df dot lth dot se ¶
17 years ago
The only way browscap examines the target browser is through the HTTP_USER_AGENT so there is no way you can determine installed plug-ins. The only way to do that is through client-side JavaScripts.

up
down
-6
Simeon ¶
6 years ago
This code is for placing a box in the corner of IE user's page warning of the disregard the web-designer has for the website's appearance in IE. Call the box with ie_box(). Style and position the box how you want using the iebox css class.

```
<?php
function using_ie()
{
    $u_agent = $_SERVER['HTTP_USER_AGENT'];
    $ub = False;
    if(preg_match('/MSIE/i',$u_agent))
    {
        $ub = True;
    }

    return $ub;
}

function ie_box() {
    if (using_ie()) {
        ?>
        <div class="iebox">
            This page is not designed for Intenet Explorer.  If you want to see this webpage as intended, please use a standards compliant browser, such as <a
href="http://www.google.com/chrome">Google Chrome</a>.
        </div>
```

```
        <?php
    return;
    }
}
?>
```

up
down
-3
Billscott ¶
2 years ago

After a great deal of research in to how to detect a users browser using php.I found out that the following code below works best so i feel like sharing.

```
<?php
$user_agent = $_SERVER['HTTP_USER_AGENT'];

if (preg_match('/safari/i', $user_agent)) {
   echo "safari";
   }else if (preg_match('/Mozilla/i', $user_agent)) {
   echo "Mozilla";
   }else if (preg_match('/chrome/i', $user_agent)) {
   echo "chrome";
   }else{
echo "You are not using any of the major browsers";
}
?>
```

up
down
-3
Mattkun Koder ¶
8 years ago

If you want to use: ceo /a/ mmg5 /./ com 's improved version and STILL detect Google Chrome you need to move CHROME earlier on the list Before Safari otherwise it will be detected as safari.

```
    $browser_list = 'msie firefox chrome konqueror safari netscape navigator opera mosaic lynx amaya omniweb avant camino flock seamonkey aol mozilla gecko';
```

up
down
-3
ansar dot ahmed at impelsys dot com ¶
10 years ago

We are using get_browser() function for useragent Mozilla/4.0 (compatible; MSIE 4.01; Windows NT) the get_browser function is returning as Default Browser and Platform = unknown.

So i added this to my browscap.ini manually:

```
[Mozilla/4.0 (compatible; MSIE 4.01; Windows NT)]
Parent=IE 4.01
Platform=WinNT
```

up
down
-3
adspeed.com ¶
12 years ago

Here is what we do to fix the parsing error messages for php_browscap.ini downloaded from Gary's website.

```
<?php
// fix the browsecap.ini for php
```

```
$v= file_get_contents('php_browscap.ini');
$v= preg_replace("/\r/","",$v);
$v= preg_replace('/="(.*)"/i','=\\1',$v);
$v= preg_replace("/platform=(.*)/i","platform=\"\\1\"",$v);
$v= preg_replace("/parent=(.*)/i","parent=\"\\1\"",$v);
$v= preg_replace("/minorver=(.*)/i","minorver=\"\\1\"",$v);
$v= preg_replace("/majorver=(.*)/i","majorver=\"\\1\"",$v);
$v= preg_replace("/version=(.*)/i","version=\"\\1\"",$v);
$v= preg_replace("/browser=(.*)/i","browser=\"\\1\"",$v);
$v= str_replace("[*]","*",$v);
file_put_contents('browscap.ini',$v);
?>
```

up
down
-2
wind_born at yahoo dot com ¶
3 years ago

Your system may have problems parsing browscap.ini if the value fields are not quoted (see bishop's note). This is often true even for the PHP specific version of the published browscap.ini. The sed command below adds quotations to the values:

```
sed '/^[a-zA-Z0-9].*=[a-zA-Z0-9]/s/=\(.*\)/="\1"/' $inFile > $fixedFile
```

If you're using a script to fetch the latest browscap.ini, consider adding this operation to the script.

up
down
-2
verx at implix dot com ¶
14 years ago

Please keep in mind that you should somehow (for example in session) cache the required results of get_browser() because it really slows thinks down.

We have experienced that without querying for browser data our scripts would run 120-130% faster. the explanation is that over 200kb long file (browscap.ini) has to be loaded and parsed everytime someone access any page (we need browser results on all pages).

So keep results in session and expect a performance boost.

up
down
-9
digibrisk at gmail dot NOSPAM dot SPAMNO dot com ¶
10 years ago

If the "browscap" directive isn't set in your server's php.ini, then an error warning is shown. Just in case, you could make a call to ini_get() to check if the browscap directive is set before using browser_get().

```
<?php
if(ini_get("browscap")) {
    $browser = get_browse(null, true);
}
?>
```

add a note