How to get the IP address in PHP?

Many times we need to get the IP address of the visitor for different purposes. It is very easy to collect the IP address in PHP. PHP provides PHP **$_SERVER** variable to get the user IP address easily. We can track the activities of the visitor on the website for the security purpose, or we can know that who uses my website and many more.

The simplest way to collect the visitor IP address in PHP is the **REMOTE_ADDR**. Pass the 'REMOTE_ADDR' in PHP $_SERVER variable. It will return the IP address of the visitor who is currently viewing the webpage.

Note: We can display this IP address on the webpage and also even can store in database for many other purposes such as - for security, redirecting a visitor to another site, blocking/banning the visitor.

Get the IP address of the website

$_SERVER['REMOTE_ADDR'] - It returns the IP address of the user currently visiting the webpage.

For example

```
<?php
echo 'User IP Address - '.$_SERVER['REMOTE_ADDR'];
?>
```

Output

```
User IP Address - ::1
```

But sometimes the REMOTE_ADDR does not return the IP address of the client, and the main reason behind is to use the proxy. In such type of situation, we will try another way to get the real IP address of the user in PHP.

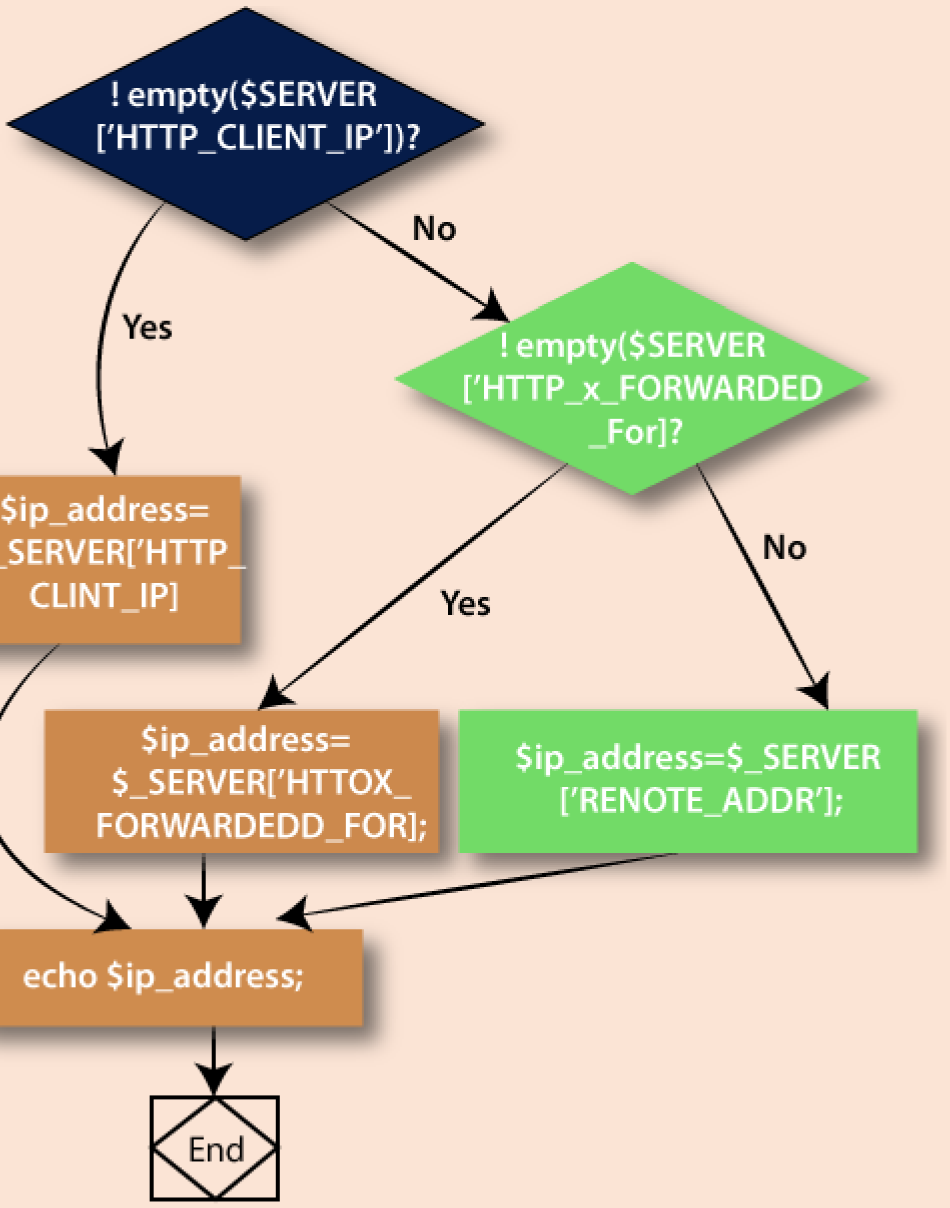
```
<?php
    function getIPAddress() {
    //whether ip is from the share internet
     if(!emptyempty($_SERVER['HTTP_CLIENT_IP'])) {
                $ip = $_SERVER['HTTP_CLIENT_IP'];
        }
    //whether ip is from the proxy
    elseif (!emptyempty($_SERVER['HTTP_X_FORWARDED_FOR'])) {
            $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
    }
//whether ip is from the remote address
    else{
            $ip = $_SERVER['REMOTE_ADDR'];
    }
     return $ip;
}
$ip = getIPAddress();
echo 'User Real IP Address - '.$ip;
?>
```

Output

```
User IP Address - ::1
```

Flowchart:

The flowchart for the above program will be like given below.


! empty($SERVER ['HTTP_CLIENT_IP'])?
No
Yes
! empty($SERVER ['HTTP_x_FORWARDED _For]?
$ip_address= $_SERVER['HTTP_ CLINT_IP]
No
Yes
$ip_address= $_SERVER['HTTOX_ FORWARDEDD_FOR];
$ip_address=$_SERVER ['RENOTE_ADDR'];
echo $ip_address;
End

Get the IP address of the website

We can also get the IP address of any website by its URL. Pass the URL of the website inside ***gethostbyname()*** function.

For example

```
<?php
$ip_address = gethostbyname("www.google.com");
echo "IP Address of Google is - ".$ip_address;
echo "</br>";
$ip_address = gethostbyname("www.javatpoint.com");
```

```
echo "IP Address of javaTpoint is - ".$ip_address;
?>
```

Output

```
IP Address of Google is - 172.217.166.4
IP Address of javaTpoint is - 95.216.57.234
```

PHP basename( ) Function

PHP functions are a self-centric block of statements that can be executed a number of times as per the developer's requirements.

PHP contains multiple inbuilt functions that can be used at times of need. In these functions, we don't have to give the definition of functions; all we have to do is declare the function and add parameters to use the same.

PHP inbuilt functions:

phpinfo()	print()	mysqli_connect()	error_reporting()
error_log()	array()	copy()	unlink()
date()	time()	strlen()	strlen()

PHP Basename() function

The basename function is an inbuilt PHP function mainly used to return the base name of a given file on a certain condition when the path of the desired file is given as a parameter inside the base name function. i.e., it gives the trailing name of the path.

Syntax:

```
1. String basename ( $ path , $ suffix )
```

PHP basename() functions hold two parameters which are mandatory for the function to execute.

Parameter	Description	Required Optional
$Path	this parameter defines the path of the file or directory, this parameter is of string type and it is mandatory to provide this parameter in order to execute the function.	Required
$Suffix	it is a non-essential parameter that is used when a file contains a name extension that end with a suffix, this parameter hides the extension	Optional

Example:

```
1)  $path = " D:\software\Autoplay\langdata\en_US/example_file.php",
2)  $path = " D:\software\Autoplay\langdata\en_US/example_file.php ",
     $suffix = ".php"
```

Exceptions:

1. Components such as ' . . ' used to get inside a file are not recognized as a path by the basename( ) function
2. The user's path is declared as a string; therefore, the basename( ) function does not recognize the actual file system used in operating systems.
3. In windows, the directory system uses forward and backward slashes (" / " , " \ ") to move in and out of the file. These are called directory separators. Still, on another operating system like Linux, we can only use a forward slash.
4. With the basename ( ) function, we can only have the base name of the declared directory whose spathe has been specified inside the basename ( ) function parameter. In order to get all the components of the file, we can use the pathinfo ( ) function, which is also an inbuilt PHP function.

Program 1:

```
<!DOCTYPE html>
<html>
<body>

<?php
$path = "D:\software\Autoplay\langdata\en_US/myfirstPHP_file.php";
echo basename($path);

?>

</body>
</html>
```

Output:

```
myfirstPHP_file.php
```

Here in this program, we have declared a variable $path and assigned the file's directory location whose base name we have to use. And in the **echo** statement, we have used the **basename ( )** function with the path as a parameter, which will let the compiler display only the file name.

Program 2:

```
<!DOCTYPE html>
<html>
<body>

<?php
$path = "D:\software\Autoplay\langdata\en_US/myfirstPHPfile.php";
echo basename($path, ".php");
?>

</body>
</html>
```

Output:

```
myfirstPHPfile
```

Here in this program, we have declared a variable **$path** and assigned the file's directory location whose base name we have to use. And in the **echo** statement, we have used the **basename ( )** function with two parameters, **path** to assign the path of file and **suffix,** which will remove the file extension, and the compiler to display only the file name without the. PHP extension.

Program 3:

```
<!DOCTYPE html>
<html>
<body>

<?php
$path = "D:\software\Autoplay\langdata\en_US/myfirstPHPfile.php";
echo " a ) ";
echo basename($path, ".php");

$path = "D:\software\Autoplay\langdata\en_US/mySecondPHPfile.php";
```

```php
echo " b ) ";
echo basename($path);

$path = "software/Autoplay/langdata/en_US";
echo " c ) ";
echo basename($path);

$path = "software/Autoplay/langdata";
echo " d ) ";
echo basename($path, ".php");

$path = "software/Autoplay";
echo " f ) ";
echo basename($path, ".php");

$path = "software";
echo " g ) ";
echo basename($path, ".php");

$path = "D:";
echo " h ) ";
echo basename($path, ".php");

$path = ".";
echo " i ) ";
echo basename($path, ".php");

$path = "/";
echo " j ) ";
echo basename($path, ".php");


$path = "D:\software\Autoplay\langdata\en_US/mySecondPHPfile.php";
echo " k ) ";
print_r(pathinfo($path));

?>
```

```

</body>
</html>
```

Output:

```
a ) myfirstPHPfile
b ) mySecondPHPfile.php
c ) en_US
d ) langdata
f ) Autoplay
g ) software
h ) D:
i ) .
j )
k ) Array (
                    [dirname] => D:\software\Autoplay\langdatan_US
                    [basename] => mySecondPHPfile.php
                    [extension] => php
                    [filename] => mySecondPHPfile )
```

Here in this program we have declared a variable **$path** and assigned multiple values to it displaying all the forms in which the base name function can be utilised.

PHP File Upload

PHP allows you to upload single and multiple files through few lines of code only.

PHP file upload features allows you to upload binary and text files both. Moreover, you can have the full control over the file to be uploaded through PHP authentication and file operation functions.

PHP $_FILES

The PHP global $_FILES contains all the information of file. By the help of $_FILES global, we can get file name, file type, file size, temp file name and errors associated with file.

Here, we are assuming that file name is *filename*.

$_FILES['filename']['name']

returns file name.

$_FILES['filename']['type']

returns MIME type of the file.

$_FILES['filename']['size']

returns size of the file (in bytes).

$_FILES['filename']['tmp_name']

returns temporary file name of the file which was stored on the server.

$_FILES['filename']['error']

returns error code associated with this file.

---

move_uploaded_file() function

The move_uploaded_file() function moves the uploaded file to a new location. The move_uploaded_file() function checks internally if the file is uploaded thorough the POST request. It moves the file if it is uploaded through the POST request.

Syntax

```
bool move_uploaded_file ( string $filename , string $destination )
```

PHP File Upload Example

File: uploadform.html

```
<form action="uploader.php" method="post" enctype="multipart/form-data">
    Select File:
    <input type="file" name="fileToUpload"/>
    <input type="submit" value="Upload Image" name="submit"/>
</form>
```

File: uploader.php

```
<?php
$target_path = "e:/";
$target_path = $target_path.basename( $_FILES['fileToUpload']['name']);
```

```php

if(move_uploaded_file($_FILES['fileToUpload']['tmp_name'], $target_path)) {
    echo "File uploaded successfully!";
} else{
    echo "Sorry, file not uploaded, please try again!";
}
?>
```

PHP Download File

PHP enables you to download file easily using built-in readfile() function. The readfile() function reads a file and writes it to the output buffer.

PHP readfile() function

Syntax

```php
int readfile ( string $filename [, bool $use_include_path = false [, resource $context ]] )
```

$filename: represents the file name

$use_include_path: it is the optional parameter. It is by default false. You can set it to true to the search the file in the included_path.

$context: represents the context stream resource.

int: it returns the number of bytes read from the file.

PHP Download File Example: Text File

File: download1.php

```php
<?php
$file_url = 'http://www.javatpoint.com/f.txt';
header('Content-Type: application/octet-stream');
header("Content-Transfer-Encoding: utf-8");
header("Content-disposition: attachment; filename=\"" . basename($file_url) . "\"");
```

```php
readfile($file_url);
?>
```

PHP Download File Example: Binary File

File: download2.php

```php
<?php
$file_url = 'http://www.myremoteserver.com/file.exe';
header('Content-Type: application/octet-stream');
header("Content-Transfer-Encoding: Binary");
header("Content-disposition: attachment; filename=\"" . basename($file_url) . "\"");
readfile($file_url);
?>
```