

[\[edit\]](#) Last updated: Fri, 27 Apr 2012

## Built-in web server

As of PHP 5.4.0, the CLI SAPI provides a built-in web server.

This web server is designed for developmental purposes only, and should not be used in production.

URI requests are served from the current working directory where PHP was started, unless the `-d` option is used to specify an explicit document root.

If a URI request does not specify a file, then either `index.php` or `index.html` in the given director are returned. If neither file exists, then a 404 response code is returned.

If a PHP file is given on the command line when the web server is started it is treated as a "router" script for the web server. The script is run at the start of each HTTP request. If this script returns **FALSE**, then the requested resource is returned as-is. Otherwise the script's output is returned to the browser.

### Example #1 Starting the web server

```
$ cd ~/public_html
$ php -S localhost:8000
```

The terminal will show:

```
PHP 5.4.0 Development Server started at Thu Jul 21 10:43:28 2011
Listening on localhost:8000
Document root is /home/me/public_html
Press Ctrl-C to quit
```

After URI requests for `http://localhost:8000/` and `http://localhost:8000/myscript.html` the terminal will show something similar to:

```
PHP 5.4.0 Development Server started at Thu Jul 21 10:43:28 2011
Listening on localhost:8000
Document root is /home/me/public_html
Press Ctrl-C to quit.
[Thu Jul 21 10:48:48 2011] ::1:39144 GET /favicon.ico - Request read
[Thu Jul 21 10:48:50 2011] ::1:39146 GET / - Request read
[Thu Jul 21 10:48:50 2011] ::1:39147 GET /favicon.ico - Request read
[Thu Jul 21 10:48:52 2011] ::1:39148 GET /myscript.html - Request read
[Thu Jul 21 10:48:52 2011] ::1:39149 GET /favicon.ico - Request read
```

### Example #2 Starting with a specific document root directory

```
$ cd ~/public_html
$ php -S localhost:8000 -t foo/
```

The terminal will show:

```
PHP 5.4.0 Development Server started at Thu Jul 21 10:50:26 2011
Listening on localhost:8000
```

```
Document root is /home/me/public_html/foo
Press Ctrl-C to quit
```

### Example #3 Using a Router Script

Requests for images will display them, but requests for HTML files will display "Welcome to PHP"

```
<?php
// router.php
if (preg_match('/\.(?:png|jpg|jpeg|gif)$/i', $_SERVER["REQUEST_URI"]))
    return false;    // serve the requested resource as-is.
else {
    echo "<p>Welcome to PHP</p>";
}
?>
```

```
$ php -S localhost:8000 router.php
```

After several URI requests the terminal will show something similar to:

```
PHP 5.4.0 Development Server started at Thu Jul 21 10:53:19 2011
Listening on localhost:8000
Document root is /home/me/public_html
Press Ctrl-C to quit.
[Thu Jul 21 10:53:45 2011] ::1:55801 GET /mylogo.jpg - Request read
[Thu Jul 21 10:53:52 2011] ::1:55803 GET /abc.html - Request read
[Thu Jul 21 10:53:52 2011] ::1:55804 GET /favicon.ico - Request read
```

#### User Contributed Notes **Built-in web server**

**necenzurat a-t gmail.com** [11-Apr-2012 11:49](#)

I am using a library as a router for php, maybe you will find this useful especially for the php webserver

<http://github.com/chriso/klein.php>

**Anonymous** [04-Apr-2012 03:09](#)

I think this has great potential for solving a lot of the Comet type issues we cant really do well in PHP. Anyone care to try some of it out?

**Anonymous** [28-Mar-2012 03:41](#)

I tried to access SVG files via the built-in webserver, but only a download dialog was shown.

So I use this router script to accomplish my needs. Maybe it is useful to someone. Bye

```
<?php
$path = pathinfo($_SERVER["SCRIPT_FILENAME"]);

if($path["extension"] == ".svg")
```

```
{
    header("Content-Type: image/svg+xml");
    readfile($_SERVER["SCRIPT_FILENAME"]);
}
else return FALSE;
?>
```

---

**gaylord at 100days dot de** [14-Mar-2012 03:47](#)

In my tests the embeeded server was only able to handle one request at a time.  
Builds a queue though, so for loading a web page it works.  
So it is really only usable for very simple test cases.

---

**Stefano F. Rausch** [09-Mar-2012 12:25](#)

To develop / deploy websites in 3 stages, i.e. ( 1 ) locally, ( 2 ) with an access controlled dedicated beta / test website in the www and ( 3 ) the production site, you can have ( 1 ) & ( 2 ) using the same domain name - port free - as follows:

- look up the IP address of the beta.web.site

and edit the hosts file to reflect:

- <IP> beta.web.site
- 127.0.0.1 beta.web.site

Start the built-in web server to work locally with:

- sudo php -S beta.web.site:80

and just hit <http://beta.web.site> as usual. Switching back and forth between ( 1 ) and ( 2 ) is as easy as telling the php engine not to fake a server any more :) Nice!

Happy PHP'ing.