

[Sign Up](#)[Sign In](#)[Search](#)

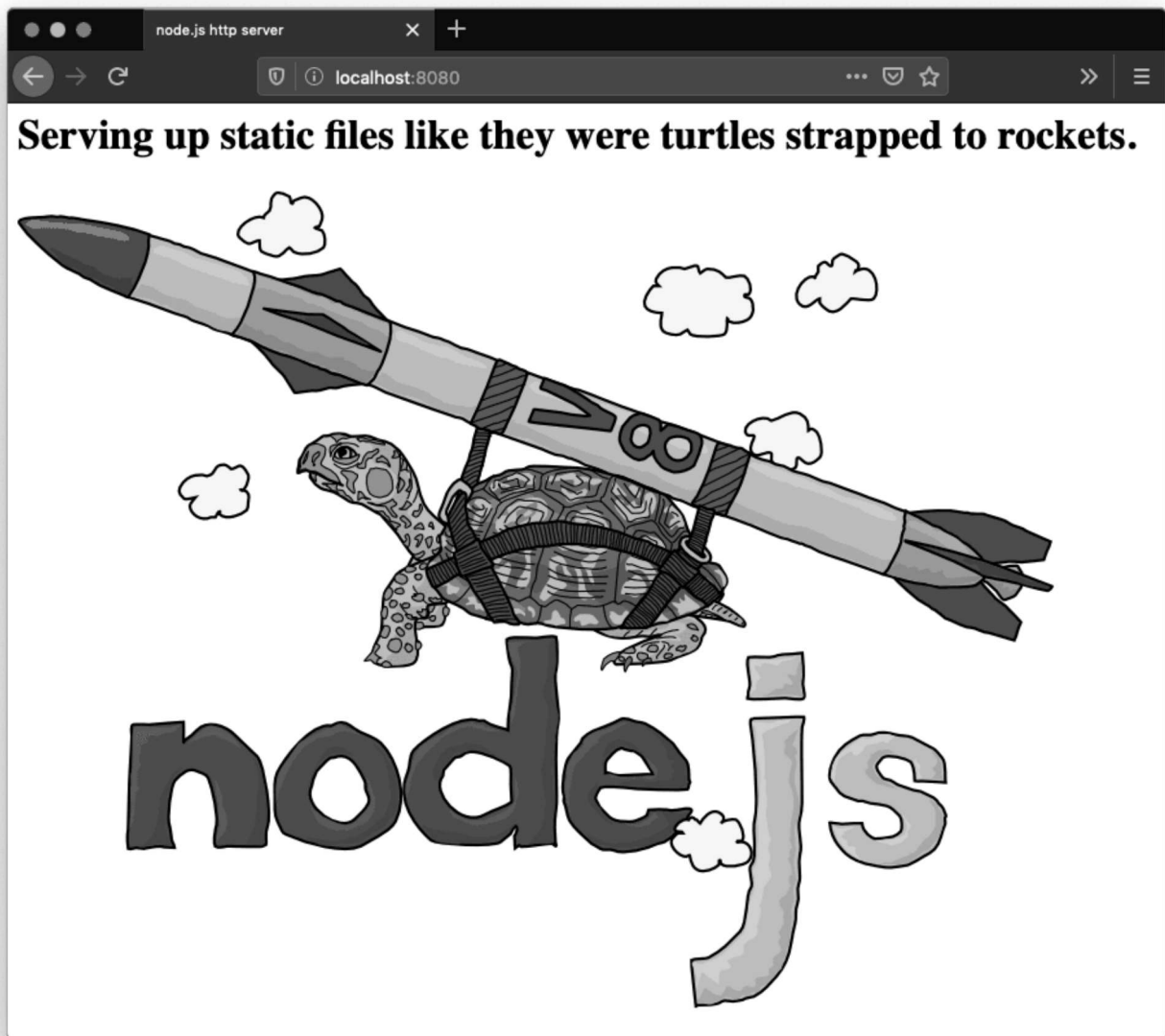
http-server DT

14.1.0 • Public • Published 2 months ago

[Readme](#)[Explore](#) BETA[13 Dependencies](#)[920 Dependents](#)[48 Versions](#)[build failing](#)[npm v14.1.0](#)[homebrew v14.1.0](#)[npm downloads 5.2M/month](#)[license MIT](#)

http-server: a simple static HTTP server

`http-server` is a simple, zero-configuration command-line static HTTP server. It is powerful enough for production usage, but it's simple and hackable enough to be used for testing, local development and learning.



Installation:

Running on-demand:

Using `npx` you can run the script without installing it first:

```
npx http-server [path] [options]
```

Globally via `npm`

```
npm install --global http-server
```

This will install `http-server` globally so that it may be run from the command line anywhere.

Globally via Homebrew

```
brew install http-server
```

As a dependency in your npm package:

```
npm install http-server
```

Usage:

```
http-server [path] [options]
```

[path] defaults to `./public` if the folder exists, and `./` otherwise.

Now you can visit ***http://localhost:8080*** to view your server

Note: Caching is on by default. Add `-c-1` as an option to disable caching.

Available Options:

Command	Description	Defaults
<code>-p</code> or <code>--port</code>	Port to use. Use <code>-p 0</code> to look for an open port, starting at 8080. It will also read from <code>process.env.PORT</code> .	8080
<code>-a</code>	Address to use	0.0.0.0
<code>-d</code>	Show directory listings	true
<code>-i</code>	Display autoIndex	true
<code>-g</code> or <code>--gzip</code>	When enabled it will serve <code>./public/some-file.js.gz</code> in place of <code>./public/some-file.js</code> when a gzipped version of the file exists and the request accepts gzip encoding. If brotli is also enabled, it will try to serve brotli first.	false

Command	Description	Defaults
<code>-b</code> or <code>--brotli</code>	When enabled it will serve <code>./public/some-file.js.br</code> in place of <code>./public/some-file.js</code> when a brotli compressed version of the file exists and the request accepts <code>br</code> encoding. If gzip is also enabled, it will try to serve brotli first.	false
<code>-e</code> or <code>--ext</code>	Default file extension if none supplied	html
<code>-s</code> or <code>--silent</code>	Suppress log messages from output	
<code>--cors</code>	Enable CORS via the <code>Access-Control-Allow-Origin</code> header	
<code>-o</code> [path]	Open browser window after starting the server. Optionally provide a URL path to open. e.g.: <code>-o /other/dir/</code>	
<code>-c</code>	Set cache time (in seconds) for cache-control max-age header, e.g. <code>-c10</code> for 10 seconds. To disable caching, use <code>-c-1</code> .	3600
<code>-U</code> or <code>--utc</code>	Use UTC time format in log messages.	
<code>--log-ip</code>	Enable logging of the client's IP address	false
<code>-P</code> or <code>--proxy</code>	Proxies all requests which can't be resolved locally to the given url. e.g.: <code>-P http://someurl.com</code>	
<code>--proxy-options</code>	Pass proxy options using nested dotted objects. e.g.: <code>--proxy-options.secure false</code>	
<code>--username</code>	Username for basic authentication	

Command	Description	Defaults
<code>--password</code>	Password for basic authentication	
<code>-S, --tls</code> or <code>--ssl</code>	Enable secure request serving with TLS/SSL (HTTPS)	<code>false</code>
<code>-C</code> or <code>--cert</code>	Path to ssl cert file	<code>cert.pem</code>
<code>-K</code> or <code>--key</code>	Path to ssl key file	<code>key.pem</code>
<code>-r</code> or <code>--robots</code>	Automatically provide a /robots.txt (The content of which defaults to <code>User-agent: *\nDisallow: /</code>)	<code>false</code>
<code>--no-dotfiles</code>	Do not show dotfiles	
<code>--mimetypes</code>	Path to a .types file for custom mimetype definition	
<code>-h</code> or <code>--help</code>	Print this list and exit.	
<code>-v</code> or <code>--version</code>	Print the version and exit.	

Magic Files

- `index.html` will be served as the default file to any directory requests.
- `404.html` will be served if a file is not found. This can be used for Single-Page App (SPA) hosting to serve the entry page.

Catch-all redirect

To implement a catch-all redirect, use the index page itself as the proxy with:


```
http-server --proxy http://localhost:8080?
```

Note the `?` at the end of the proxy URL. Thanks to [@houston3](#) for this clever hack!

TLS/SSL

First, you need to make sure that **openssl** is installed correctly, and you have `key.pem` and `cert.pem` files. You can generate them using this command:

```
openssl req -newkey rsa:2048 -new -nodes -x509 -days 3650 -keyout ke
```



You will be prompted with a few questions after entering the command. Use `127.0.0.1` as value for `Common name` if you want to be able to install the certificate in your OS's root certificate store or browser so that it is trusted.

This generates a cert-key pair and it will be valid for 3650 days (about 10 years).

Then you need to run the server with `-S` for enabling SSL and `-C` for your certificate file.

```
http-server -S -C cert.pem
```

If you wish to use a passphrase with your private key you can include one in the `openssl` command via the `-passout` parameter (using password of foobar)

```
e.g. openssl req -newkey rsa:2048 -passout pass:foobar -keyout key.pem  
-x509 -days 365 -out cert.pem
```

For security reasons, the passphrase will only be read from the `NODE_HTTP_SERVER_SSL_PASSPHRASE` environment variable.

This is what should be output if successful:

```
Starting up http-server, serving ./ through https
```

```
http-server settings:
```

CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:

`https://127.0.0.1:8080`

`https://192.168.1.101:8080`

`https://192.168.1.104:8080`

Hit CTRL-C to stop the server

Development

Checkout this repository locally, then:

```
$ npm i
```

```
$ npm start
```

*Now you can visit **`http://localhost:8080`** to view your server*

You should see the turtle image in the screenshot above hosted at that URL. See the `./public` folder for demo content.

Keywords

`cli` `command` `static` `http` `https` `http-server` `https-server` `server`

Install

```
> npm i http-server
```

Repository

 github.com/http-party/http-server

Homepage

 github.com/http-party/http-server#readme

↓ Weekly Downloads

1 163 807



Version

14.1.0

License

MIT

Unpacked Size

124 kB

Total Files

20

Issues

65

Pull Requests

28

Last publish

2 months ago

Collaborators



 [Try on RunKit](#)

 [Report malware](#)



Support

[Help](#)

[Advisories](#)

[Status](#)

[Contact npm](#)

Company

[About](#)

[Blog](#)

[Press](#)

Terms & Policies

[Policies](#)

[Terms of Use](#)

[Code of Conduct](#)

[Privacy](#)