

---

# **wsgi-proxy Documentation**

*Release 0.4.0*

**Mikeal Rogers, Hong Minhee**

**Aug 28, 2017**



---

## Contents

---

<b>1</b>	<b>Installtion</b>	<b>3</b>
1.1	The latest release . . . . .	3
1.2	Bleeding edge . . . . .	3
1.3	For contribution . . . . .	3
<b>2</b>	<b>Running</b>	<b>5</b>
2.1	<code>wsgi-proxy</code> command . . . . .	5
2.2	WSGI application . . . . .	5
<b>3</b>	<b>Using</b>	<b>7</b>
3.1	Mac OS X . . . . .	7
3.2	Mozilla Firefox . . . . .	8
3.3	Opera . . . . .	9
3.4	<code>http_proxy</code> . . . . .	9
3.5	GNU Wget . . . . .	9
3.6	<code>curl</code> . . . . .	9
<b>4</b>	<b><code>wsgi_proxy</code> — Python package of <code>wsgi-proxy</code></b>	<b>11</b>
<b>5</b>	<b><code>wsgi_proxy.cli</code> — <code>wsgi-proxy</code> command</b>	<b>13</b>
<b>6</b>	<b><code>wsgi_proxy.version</code> — Version data</b>	<b>15</b>
<b>7</b>	<b>Changelog</b>	<b>17</b>
7.1	Version 0.4.0 . . . . .	17
7.2	Version 0.3.1 . . . . .	17
7.3	Version 0.3.0 . . . . .	17
<b>8</b>	<b>Repository and bug tracker</b>	<b>19</b>
<b>9</b>	<b>Author and license</b>	<b>21</b>
	<b>Python Module Index</b>	<b>23</b>



This package implements a simple WSGI app that implements HTTP 1.0 proxy.



### The latest release

The easiest way to install `wsgi-proxy` is to use `pip` or `easy_install`:

```
$ pip install wsgi-proxy
```

### Bleeding edge

You can install it from the repository if you use `pip`:

```
$ pip install hg+https://bitbucket.org/dahlia/wsgi-proxy
```

### For contribution

If you want to contribute to the project, you should clone the repository first. We use [Mercurial](#).

```
$ hg clone https://bitbucket.org/dahlia/wsgi-proxy
```

`pip` can install the package as editable mode through `-e` option. It just makes a link to the working directory in `site-packages`.

```
$ cd wsgi-proxy/  
$ pip install -e .
```

Or you can use `develop` command `setup.py` script provides:

```
$ cd wsgi-proxy/  
$ python setup.py develop
```





### wsgi-proxy command

The package also installs **wsgi-proxy** command on your system. It runs a proxy server on your system.

```
$ wsgi-proxy -p 8080
```

You can change the server implementation using `--server` option. Default is `waitress`.

```
$ wsgi-proxy --server wsgiref
$ wsgi-proxy --server cherrypy
```

### WSGI application

The package provides a WSGI app that implements HTTP proxy as its name says. You can serve the application using any WSGI servers like [Green Unicorn](#) or [Tornado](#). The application endpoint is:

`wsgi_proxy.app` (or some servers accept `wsgi_proxy:app`).

The following list shows some examples:

#### Green Unicorn

```
$ pip install gunicorn
$ gunicorn wsgi_proxy:app
```

#### Tornado

```
$ pip install tornado
```

```
from tornado.httpserver import HTTPServer
from tornado.ioloop import IOLoop
from tornado.wsgi import WSGIContainer
```

```
from wsgi_proxy import app

container = WSGIContainer(app)
http_server = HTTPServer(container)
http_server.listen(8080)
IOLoop.instance().start()
```

**See also:**

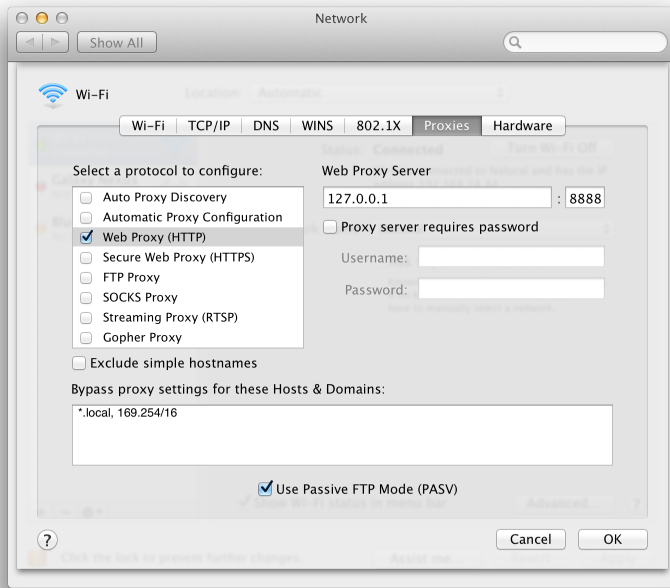
**Servers which support WSGI** — **WSGI.org** An alphabetic list of WSGI servers.

Suppose **wsgi-proxy** is listening 127.0.0.1:8888. You can set it using `--host` and `--port` options:

```
$ wsgi-proxy --host 127.0.0.1 --port 8888
```

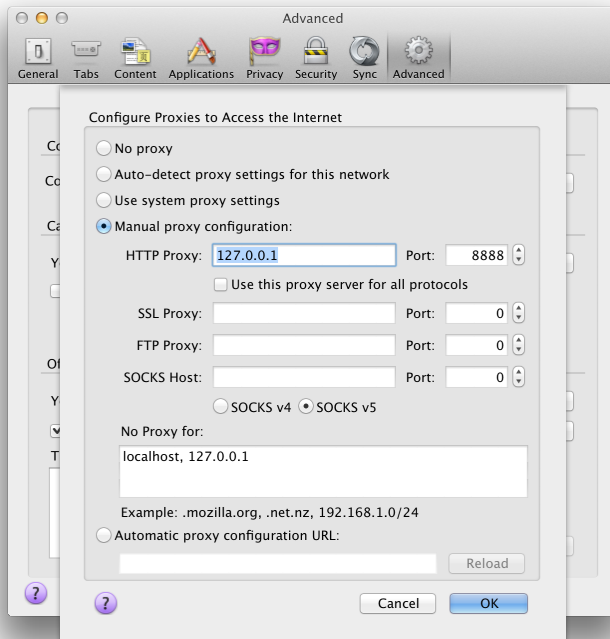
## Mac OS X

There is system-wide settings for HTTP proxy in Mac OS X. It affects to Safari and Google Chrome. You can find it on *System Preferences...* → *Network* → *Advanced...* → *Proxies*. Switch on *Web Proxy (HTTP)*, and then fill host and port number of *Web Proxy Server* to 127.0.0.1 and 8888.



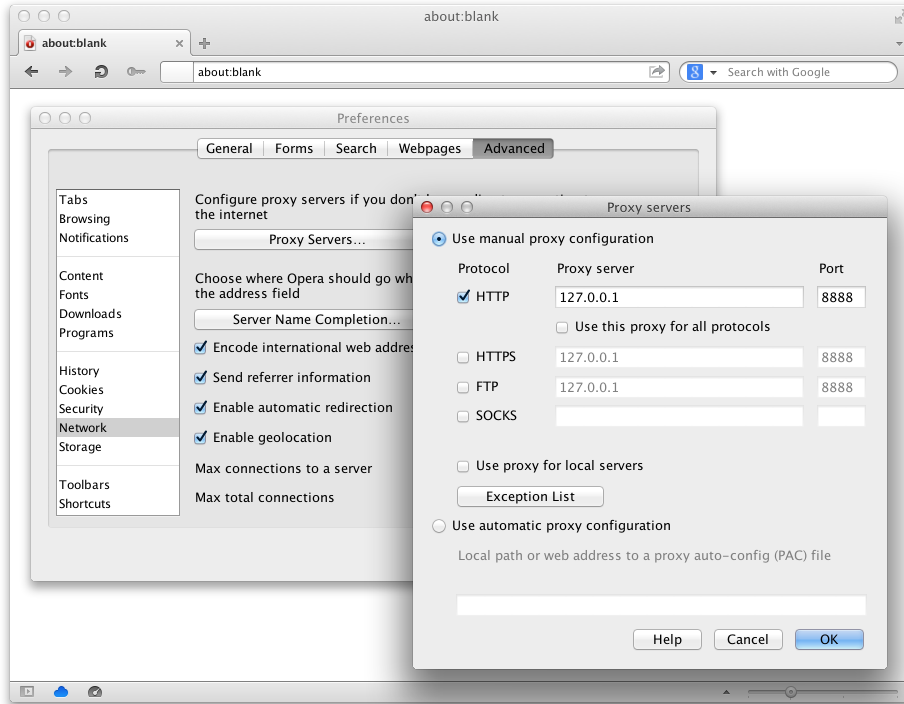
## Mozilla Firefox

Firefox provides its own proxy settings. You can find it on *Preferences...* → *Advanced* → *Network* → *Connection* → *Settings...*. Choose *Manual proxy configuration*, and then fill *HTTP Proxy* to 127.0.0.1 and its *Port* to 8888.



## Opera

Opera provides its own proxy settings. You can find it on *Preferences... → Advanced → Network → Proxy Servers...* Choose *Use manual proxy configuration*, and then fill *Proxy server* and *Port of HTTP* to 127.0.0.1 and 8888.



## http\_proxy

A lot of Unix utilities e.g. **curl**, **wget** and networking libraries e.g. **urllib2** respect `http_proxy` environment variable.

```
$ http_proxy="127.0.0.1:8888" curl http://python.org/
$ http_proxy="127.0.0.1:8888" wget http://python.org/
```

## GNU Wget

**wget** command respects `http_proxy` environment variable:

```
$ http_proxy="127.0.0.1:8888" wget http://python.org/
```

## curl

**curl** command provides `--proxy1.0` option:

```
$ curl --proxy1.0 127.0.0.1:8888 http://python.org/
```

It respects `http_proxy` environment variable as well:

```
$ http_proxy="127.0.0.1:8888" curl http://python.org/
```

---

## wsgi\_proxy — Python package of `wsgi-proxy`

---

This is an importable Python package that consists of basic WSGI app and other frontends around it. The most important thing for you is probably `app` and its type `WSGIProxyApplication`.

You can use this with other third-party WSGI servers. For example:

```
$ pip install gunicorn
$ gunicorn -p8080 wsgi_proxy:app
```

**class** `wsgi_proxy.WSGIProxyApplication`

WSGI application to handle requests that need to be proxied. You have to instantiate the class before using it as WSGI app:

```
from wsgiref.simple_server import make_server

app = WSGIProxyApplication()
make_server('', 8080, app).serve_forever()
```

**handler** (*environ*, *start\_response*)

Proxy for requests to the actual http server

`wsgi_proxy.is_hop_by_hop` (*header*)

Returns True if the given header is hop by hop.

**Parameters** **header** (basestring) – the header name

**Returns** whether the given header is hop by hop or not

**Return type** `bool`

`wsgi_proxy.reconstruct_url` (*environ*)

Reconstruct the remote url from the given WSGI environ dictionary.

**Parameters** **environ** (`collections.MutableMapping`) – the WSGI environment

**Returns** the remote url to proxy

**Return type** basestring





---

## wsgi\_proxy.cli — wsgi-proxy command

---

- h, --help**  
Print help message.
- p <port>, --port <port>**  
The port number to listen. Default is 8080.
- H <host>, --host <host>**  
The hostname to listen. Default is 127.0.0.1.
- server <server>**  
The WSGI server implementation to use. Default is waitress.
- v, --verbose**  
Print debug logs as well. It internally sets `logging` level to DEBUG.
- q, --quiet**  
Operate quietly. It internally sets `logging` level to ERROR.



## CHAPTER 6

---

`wsgi_proxy.version` — Version data

---



### Version 0.4.0

Released on August 29, 2017.

- Python 3 support. [PR #1 by Jasper Spaans]

### Version 0.3.1

Released on January 30, 2015.

- Fix a url reconstruction bug on several WSGI servers like [Green Unicorn](#). Now it sees `HTTP_HOST` and `wsgi.url_scheme` when `PATH_INFO` is not enough. [issue #1]

### Version 0.3.0

Released on April 5, 2013. It's a first version released by [Hong Minhee](#), a new maintainer of **wsgi-proxy**.

- Introduce new **wsgi-proxy** command.
- Use `waitress` by default.
- Add `wsgi_proxy.app`, a default instance of `wsgi_proxy.WSGIProxyApplication`.
- Fix `reconstruct_url()` to correctly constructs the remote URL. [#1 from Google Code]



---

### Repository and bug tracker

---

This project is currently hosted on Bitbucket. We use [Mercurial](#) as SCM.

<https://bitbucket.org/dahlia/wsgi-proxy>

Issue tracker can be find from the following link:

<https://bitbucket.org/dahlia/wsgi-proxy/issues>

Feel free to report bugs or suggest improvements.





## CHAPTER 9

---

### Author and license

---

It was originally written by [Mikeal Rogers](#), and then has been maintained by [Hong Minhee](#). Distributed under [Apache License 2.0](#).



**W**

wsgi\_proxy, 10

wsgi\_proxy.cli, 11

wsgi\_proxy.version, 13



## Symbols

`-server <server>`  
wsgi-proxy command line option, 13

`-H <host>, -host <host>`  
wsgi-proxy command line option, 13

`-h, -help`  
wsgi-proxy command line option, 13

`-p <port>, -port <port>`  
wsgi-proxy command line option, 13

`-q, -quiet`  
wsgi-proxy command line option, 13

`-v, -verbose`  
wsgi-proxy command line option, 13

## E

environment variable  
http\_proxy, 9, 10

## H

handler() (wsgi\_proxy.WSGIProxyApplication method),  
11

http\_proxy, 9, 10

## I

is\_hop\_by\_hop() (in module wsgi\_proxy), 11

## R

reconstruct\_url() (in module wsgi\_proxy), 11

## W

wsgi-proxy command line option  
-server <server>, 13  
-H <host>, -host <host>, 13  
-h, -help, 13  
-p <port>, -port <port>, 13  
-q, -quiet, 13  
-v, -verbose, 13

wsgi\_proxy (module), 10

wsgi\_proxy.cli (module), 11

wsgi\_proxy.version (module), 13

WSGIProxyApplication (class in wsgi\_proxy), 11