
wsgi-proxy Documentation

Release 0.3.1

Mikeal Rogers, Hong Minhee

May 13, 2016

1	Installation	3
1.1	The latest release	3
1.2	Bleeding edge	3
1.3	For contribution	3
2	Running	5
2.1	<code>wsgi-proxy</code> command	5
2.2	WSGI application	5
3	Using	7
3.1	Mac OS X	7
3.2	Mozilla Firefox	8
3.3	Opera	9
3.4	<code>http_proxy</code>	10
3.5	GNU Wget	10
3.6	<code>curl</code>	10
4	<code>wsgi_proxy</code> — Python package of <code>wsgi-proxy</code>	13
5	<code>wsgi_proxy.cli</code> — <code>wsgi-proxy</code> command	15
6	<code>wsgi_proxy.version</code> — Version data	17
7	Changelog	19
7.1	Version 0.3.1	19
7.2	Version 0.3.0	19
8	Repository and bug tracker	21
9	Author and license	23
	Python Module Index	25

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

Installation

1.1 The latest release

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

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

1.2 Bleeding edge

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

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

1.3 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
```


2.1 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
```

2.2 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.

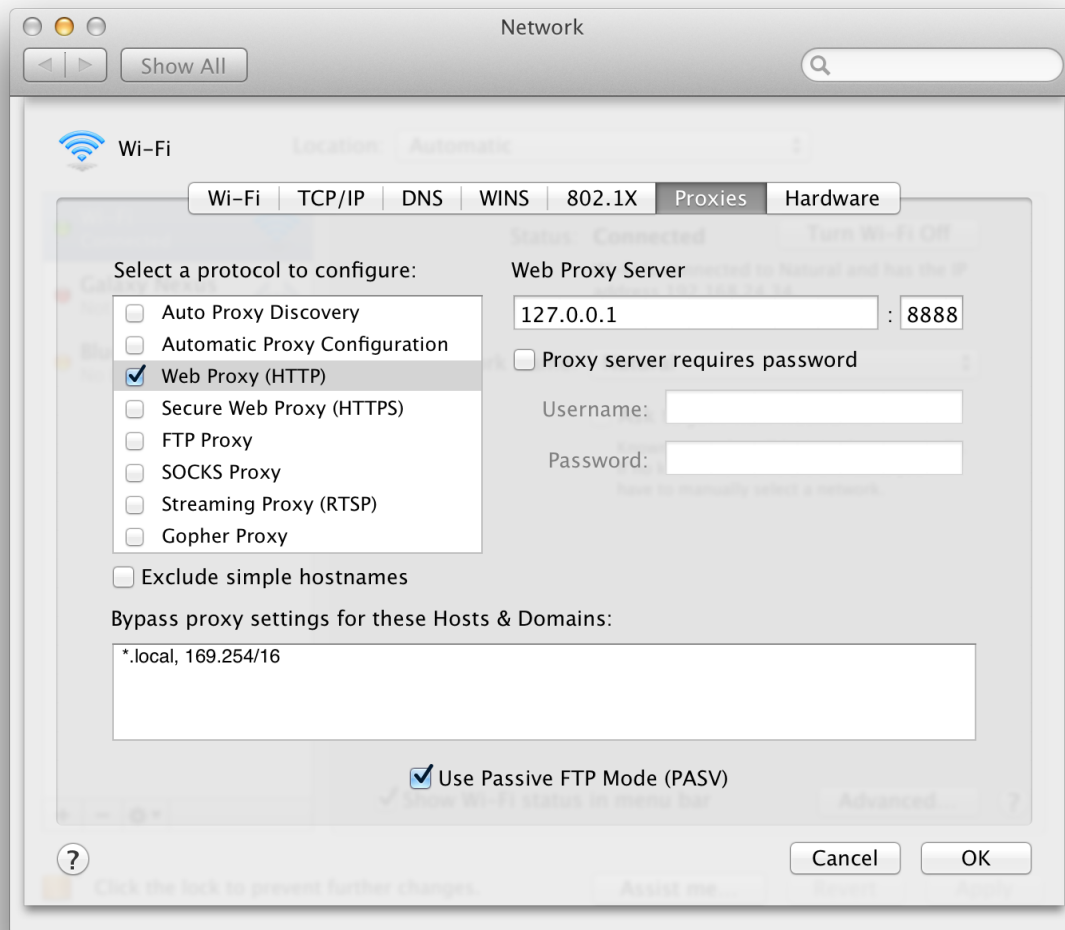
Using

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
```

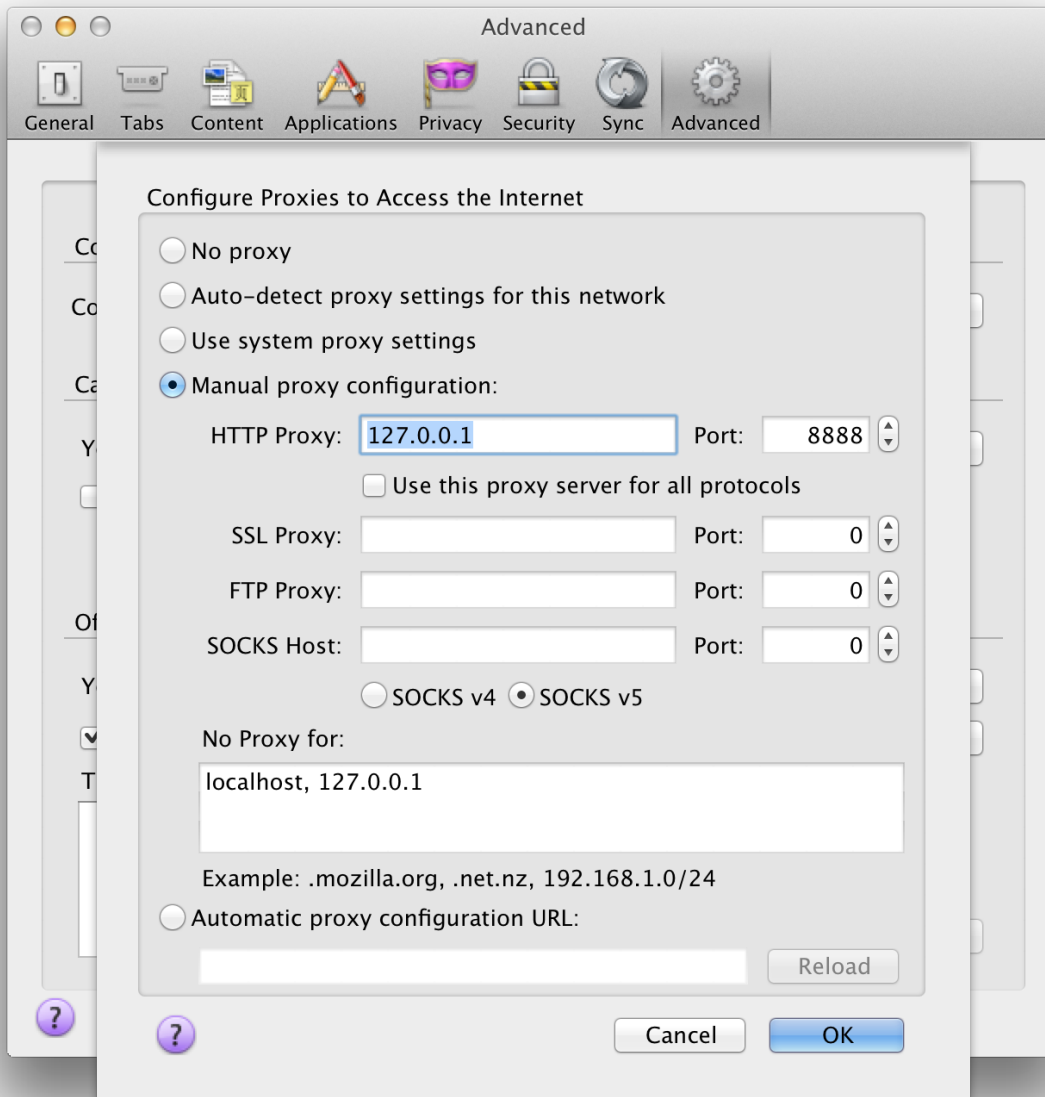
3.1 Mac OS X

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.



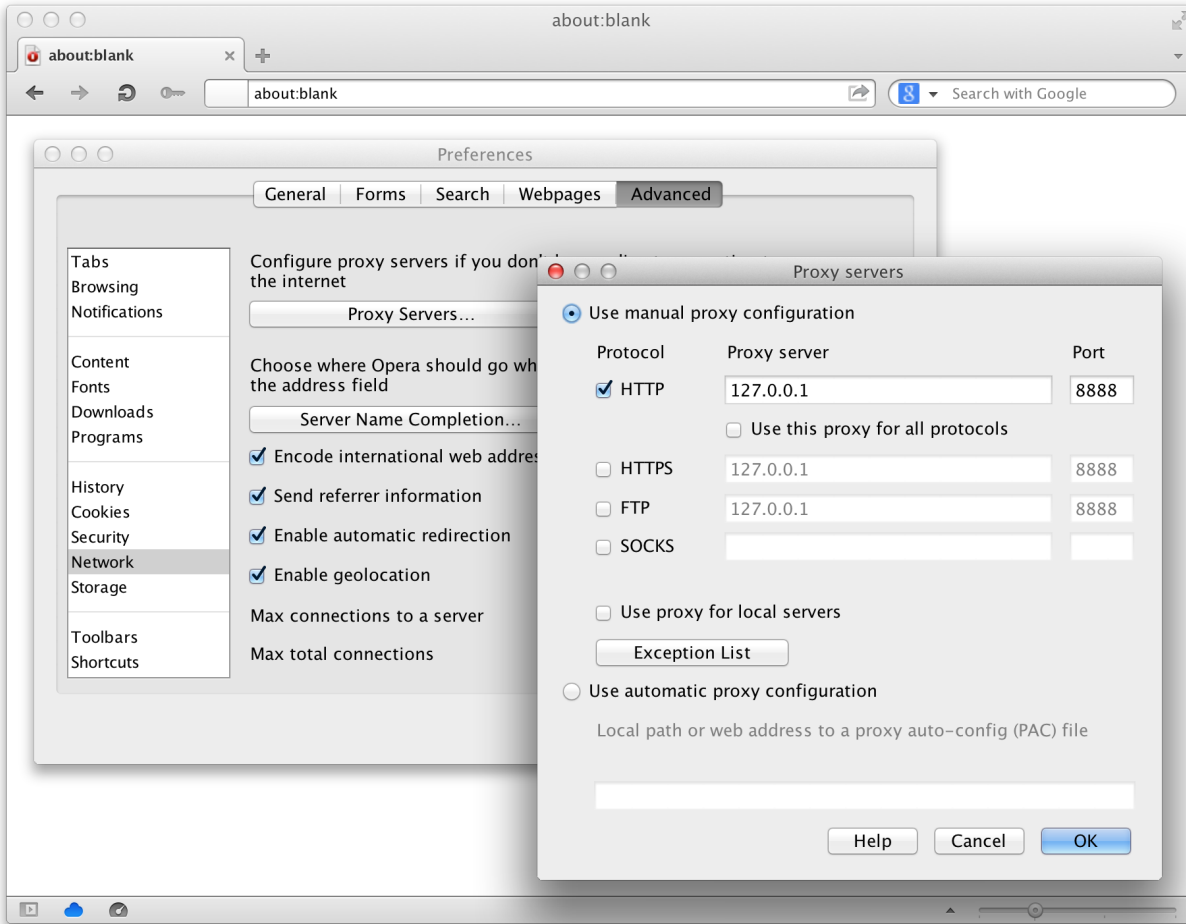
3.2 Mozilla Firefox

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.



3.3 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.



3.4 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/
```

3.5 GNU Wget

wget command respects `http_proxy` environment variable:

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

3.6 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`.

`wsgi_proxy.version` — Version data

Changelog

7.1 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]

7.2 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.

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`, 11

`wsgi_proxy.cli`, 13

`wsgi_proxy.version`, 15

Symbols

-server <server>
wsgi-proxy command line option, 15

-H <host>, -host <host>
wsgi-proxy command line option, 15

-h, -help
wsgi-proxy command line option, 15

-p <port>, -port <port>
wsgi-proxy command line option, 15

-q, -quiet
wsgi-proxy command line option, 15

-v, -verbose
wsgi-proxy command line option, 15

E

environment variable
http_proxy, 10, 11

H

handler() (wsgi_proxy.WSGIProxyApplication method),
13

http_proxy, 10, 11

I

is_hop_by_hop() (in module wsgi_proxy), 13

R

reconstruct_url() (in module wsgi_proxy), 13

W

wsgi-proxy command line option
-server <server>, 15
-H <host>, -host <host>, 15
-h, -help, 15
-p <port>, -port <port>, 15
-q, -quiet, 15
-v, -verbose, 15

wsgi_proxy (module), 11

wsgi_proxy.cli (module), 13

wsgi_proxy.version (module), 15

WSGIProxyApplication (class in wsgi_proxy), 13