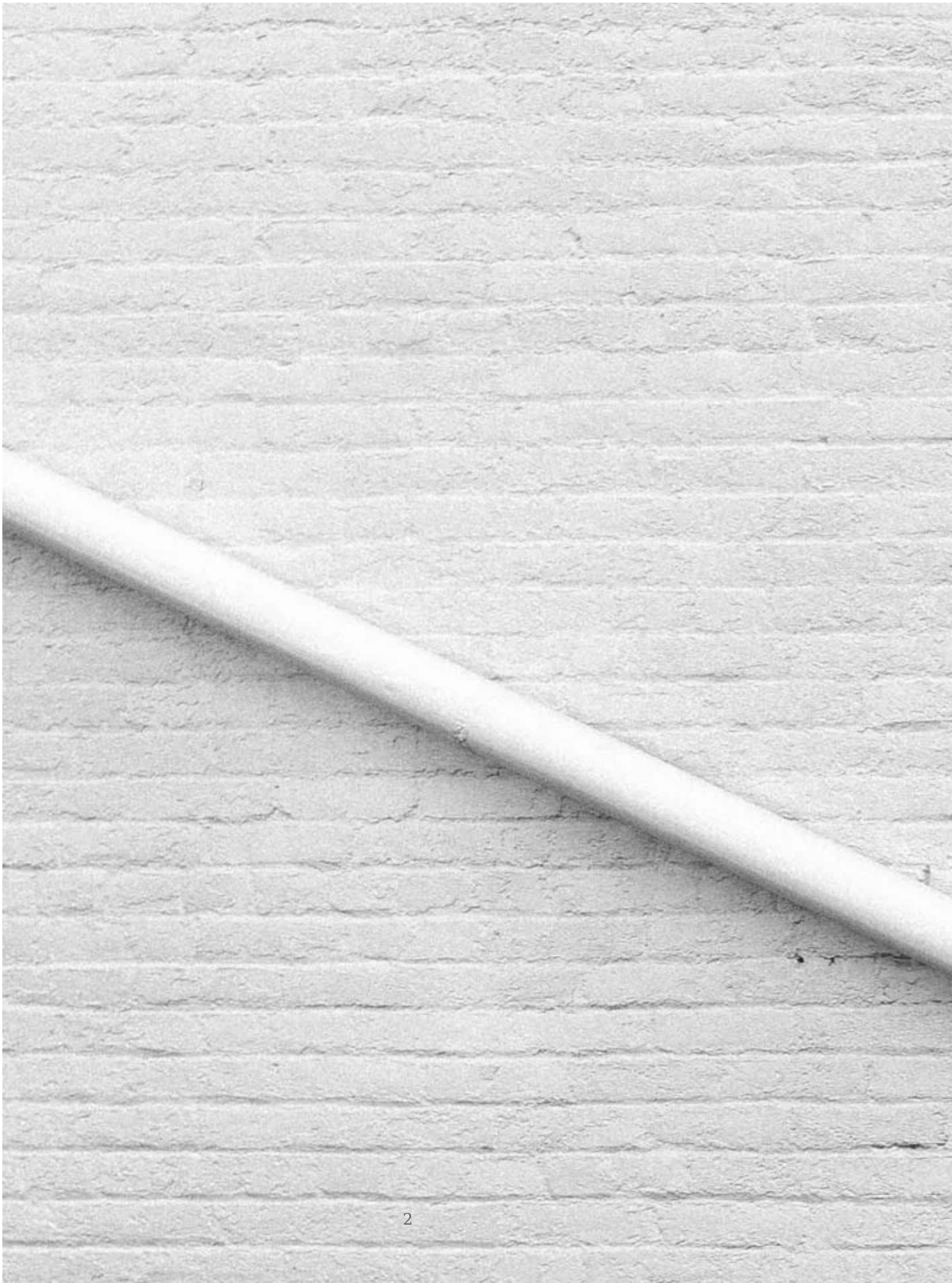
Introducing Flask-Sockets

2013

1 min read • 211 words



The state of WebSockets in Python is unfortunate — there's no obvious way to do it. Twisted + Autobhan? Node.js + HAProxy? Diesel.io? Nothing feels right. Let's create a WebSocket echo endpoint.

```
from flask import Flask
from flask_sockets import Sockets

app = Flask(__name__)
sockets = Sockets(app)

@sockets.route('/echo')
def echo_socket(ws):
    while True:
        message = ws.receive()
        ws.send(message)

@app.route('/')
def hello():
    return 'Hello World!'
```

Serving WebSockets in Python was really difficult. Now it's not.

> I'm going to use the shit out of this. — Randall Degges
>
> This looks absolutely incredible. — Glenn Siegman
>
> How do you install this in node? — Nick Hudkins
>
> You are a golden god, sir. — Jeremy Bowers
>
> *foams at the mouth* — Kyle Conroy

Installation & Deployment

Flask-Sockets is an easy to install Flask extension:

```
$ pip install Flask-Sockets
```

Production services are provided by gevent and gevent-websocket. Anything that inserts `wsgi.websocket` into the WSGI environ is supported, but gevent-websocket is recommended.

A custom Gunicorn worker is included to make deployment as friendly as possible:

```
$ gunicorn -k flask_sockets.worker hello:app
```

Everything else is taken care of for you.

Moving Forward

If you'd like to help bring this library to the next level, fork it and send a pull request!

Related Links

- Flask-Sockets on PyPi
- Flask-Sockets on GitHub
- Gevent-WebSocket
- RFC 6455

Generated from kennethreitz.org • 2025