
Spa Documentation

Release 0.10.1

Brent Tubbs

May 23, 2017

Contents

1 Quickstart

3

Spa is a Python micro-framework for building REST APIs and single-page-applications. It supports APIs implementing the RESTSockets pattern (streaming updates to API resources over WebSockets).

Here's a simple Spa app:

```
from gevent.monkey import patch_all; patch_all()

import time
from datetime import datetime

import spa

class Hello(spa.Handler):
    def get(self, name=None):
        if name:
            return spa.Response('hello ' + name)
        return spa.Response('hello!')

    def websocket(self, name=None):
        if name:
            self.ws.send('hello ' + name)

        while True:
            name = self.ws.receive()
            if name:
                self.ws.send('hello ' + name)

class DateTime(spa.Handler):
    def get(self):
        return spa.Response(datetime.now().isoformat())

    def websocket(self):
        while True:
            self.ws.send(datetime.now().isoformat())
            time.sleep(1)
```

```
routes = (  
    ('/', 'home', Hello),  
    ('/hello/<name>/', 'hello', Hello),  
    ('/datetime/', 'datetime', DateTime),  
)  
  
app = spa.App(routes)  
spa.run(app)
```

The `DateTime` handler demonstrates the `RESTSockets` pattern at work. When making an HTTP GET request, the client receives the current value. When making a WebSocket connection, the client receives a stream of values over time. You can test this locally with the `wscat` utility:

```
wscat -c ws://localhost:8000/datetime/
```

Contents: