
FigureStream

Yeison Cardona

May 26, 2021

CONTENTS

1	Instalation	3
2	Bare minimum	5
3	Set host, port and endpoint	7
4	Indices and tables	9

A backend for serve Matplotlib animations as web streams.

INSTALATION

```
pip install figurestream
```


BARE MINIMUM

By default, the stream serves on `http://localhost:5000`

```
# FigureStream replace any Figure object
from figurestream import FigureStream

import numpy as np
from datetime import datetime

# FigureStream can be used like any Figure object
stream = FigureStream()
sub = stream.add_subplot(111)
x = np.linspace(0, 3, 1000)

# Update animation loop
while True:
    sub.clear() # clear the canvas

    # -----
    # Any plot operation
    sub.set_title('FigureStream')
    sub.set_xlabel('Time [s]')
    sub.set_ylabel('Amplitude')
    sub.plot(x, np.sin(2 * np.pi * 2 * (x + datetime.now().timestamp()))))
    sub.plot(x, np.sin(2 * np.pi * 0.5 * (x + datetime.now().timestamp()))))
    # -----

    stream.feed() # push the frame into the server
```

For fast updates is recommended to use `set_data`, `set_ydata` and `set_xdata` instead of `clear` and draw again in each loop, also `FigureStream` can be implemented from a custom class.

```
# FigureStream replace any Figure object
from figurestream import FigureStream

import numpy as np
from datetime import datetime

class FastAnimation(FigureStream):

    def __init__(self, *args, **kwargs):
```

(continues on next page)

```

super().__init__(*args, **kwargs)

axis = self.add_subplot(111)
self.x = np.linspace(0, 3, 1000)

# -----
# Single time plot configuration
axis.set_title('FigureStream')
axis.set_xlabel('Time [s]')
axis.set_ylabel('Amplitude')

axis.set_ylim(-1.2, 1.2)
axis.set_xlim(0, 3)

# Lines objects
self.line1, _ = axis.plot(self.x, np.zeros(self.x.size))
self.line2, _ = axis.plot(self.x, np.zeros(self.x.size))
# -----

self.anim()

def anim(self):
    # Update animation loop
    while True:
        # -----
        # Update only the data values is faster than update all the plot
        self.line1.set_ydata(np.sin(2 * np.pi * 2 * (self.x + datetime.now().
↪timestamp()))))
        self.line2.set_ydata(np.sin(2 * np.pi * 0.5 * (self.x + datetime.now().
↪timestamp()))))
        # -----

        self.feed() # push the frame into the server

if __name__ == '__main__':
    FastAnimation()

```

SET HOST, PORT AND ENDPOINT

If we want to serve the stream in a different place we can use the parameters `host`, `port` and `endpoint`, for example:

```
FigureStream(host='0.0.0.0', port='5500', endpoint='figure.jpeg')
```

Now the stream will serve on <http://localhost:5500/figure.jpeg> and due the `0.0.0.0` host is accesible for any device on network.

By default host is `localhost`, port is `5000` and endpoint is empty.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`