Timeflux web app engine

Release 0.5.3.dev2+g955a38a

Pierre Clisson

Jun 04, 2023

CONTENTS

This plugin provides a framework to develop web applications that can interface with Timeflux. A monitoring web interface is bundled. It is not feature-complete yet, but it already allows you to visualize your data streams in real-time, in your browser, and to send events. Other examples include a SSVEP scheduler and a P300 speller. Both can be found in the `apps` directory.

CHAPTER

ONE

INSTALLATION

First, make sure that Timeflux is installed.

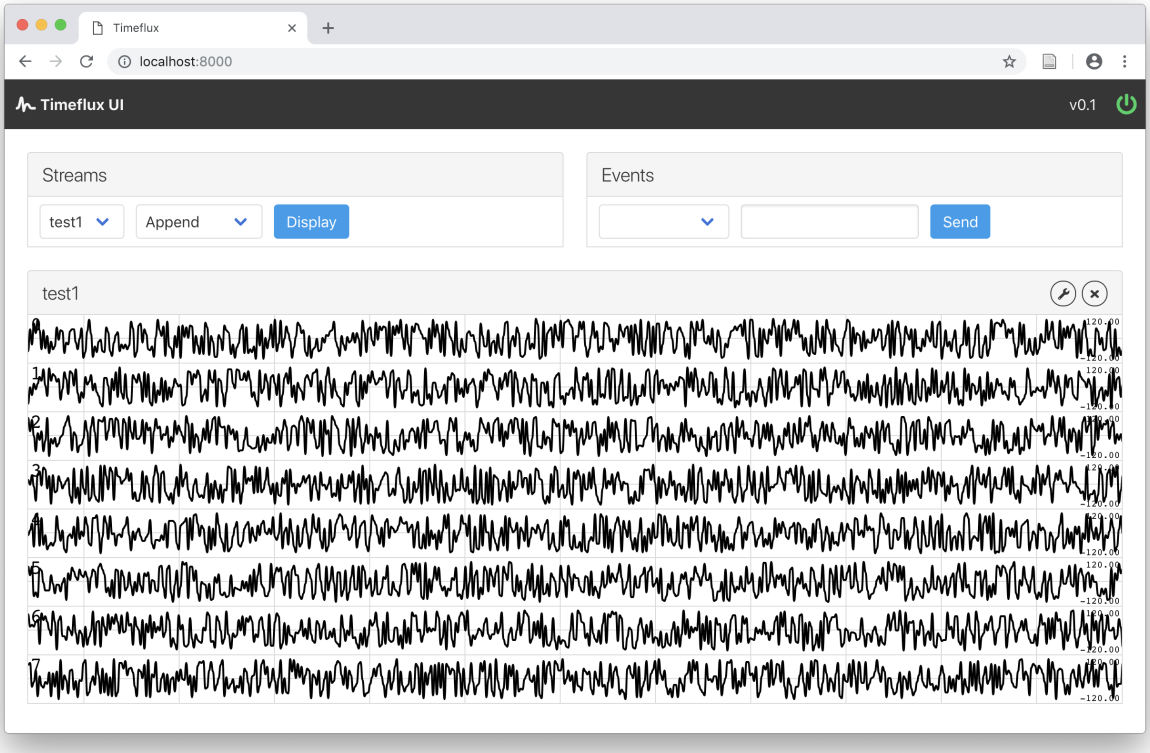
You can then install this plugin in the *timeflux* environment:

```
$ conda activate timeflux
$ pip install timeflux_ui
```

CHAPTER

TWO

SCREENSHOT



2.1 API Reference

This page contains auto-generated API reference documentation.

timeflux_ui

2.1.1 timeflux_ui

timeflux_ui.nodes

nodes

timeflux_ui.nodes.ui

ui

class timeflux_ui.nodes.ui.**UI**(*host='localhost'*, *port=8000*, *routes={}*, *settings={}*, *debug=False*)

Bases: timeflux.core.node.Node

Interact with Timeflux from the browser.

This node provides a web interface, available at `http://localhost:8000` by default. Bi-directional communication is available through the WebSocket protocol.

A real-time data stream visualization application is provided at `http://localhost:8000/monitor/`. Other example applications (such as P300 and EEG signal quality) are provided in the `apps` directory of this package.

This node accepts any number of named input ports. Streams received from the browser are forwarded to output ports.

Variables

- **i_*** (*Port*) – Dynamic inputs, expect DataFrame.
- **o_*** (*Port*) – Dynamic outputs, provide DataFrame.

Example

```
graphs:

  - nodes:
    - id: data1
      module: timeflux.nodes.random
      class: Random
      params:
        columns: 8
        rows_min: 10
        rows_max: 10
        value_min: -100
        value_max: 100
        seed: 1
    - id: data2
      module: timeflux.nodes.random
      class: Random
      params:
        columns: 2
```

(continues on next page)

(continued from previous page)

```
      rows_min: 1
      rows_max: 1
      value_min: -100
      value_max: 100
      seed: 1
  - id: ui
    module: timeflux_ui.nodes.ui
    class: UI
  - id: events
    module: timeflux.nodes.debug
    class: Display
  - id: test
    module: timeflux.nodes.debug
    class: Display
  edges:
  - source: data1
    target: ui:test1
  - source: data2
    target: ui:test2
  - source: ui:events
    target: events
  - source: data1
    target: test
  rate: 5
```

Parameters

- **host** (*string*) – The host to bind to.
- **port** (*int*) – The port to listen to.
- **routes** (*dict*) – A dictionary of custom web apps. Key is the name, value is the path.
- **settings** (*dict*) – An arbitrary configuration file that will be exposed to web apps.
- **debug** (*bool*) – Show dependencies debug information.

update()

Update the input and output ports.

terminate()

Perform cleanup upon termination.

PYTHON MODULE INDEX

t

INDEX

M

T

U