Cyclotron Documentation

Release 1.3.0

R. Picard

Oct 19, 2021

Contents:

Cyclotron is a functional and reactive framework for python, asyncio, and RxPY. It is inspired from the CycleJs javascript framework. Cyclotron makes it easy to write asynchronous reactive applications in a functional way.

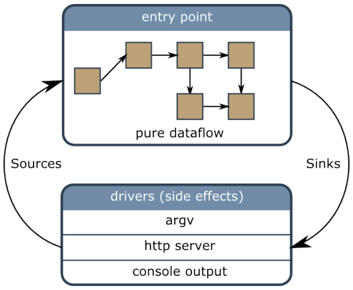
CHAPTER 1

Rationale

A Cyclotron application is composed of two parts:

- Components, that are composed of pure code.
- Drivers, that implement side effects.

Components and drivers comunicate via Observables. Communication between components and drivers is done via sink Observables. Communication between drivers and components is done via source Observables.



CHAPTER 2

Get Started

install cyclotron asyncio package:

```
$ pip3 install cyclotron-aiohttp
```

```
from collections import namedtuple

from cyclotron import Component
from cyclotron.asyncio.runner import run
import cyclotron_aiohttp.httpd as httpd
import rx
import rx.operators as ops

EchoSource = namedtuple('EchoSource', ['httpd'])
EchoSink = namedtuple('EchoSink', ['httpd'])
EchoDrivers = namedtuple('EchoDrivers', ['httpd'])

def echo_server(source):
    init = rx.from_([
        httpd.Initialize(),
        httpd.AddRoute(methods=['GET'], path='/echo/{what}', id='echo'),
        httpd.StartServer(host='localhost', port=8080),
    ])

    echo = source.httpd.route.pipe(
        ops.filter(lambda i: i.id == 'echo'),
        ops.flat_map(lambda i: i.request),
        ops.map(lambda i: httpd.Response(
            context=i.context,
            data=i.match_info['what'].encode('utf-8')),
        )
    )

    control = rx.merge(init, echo)
    return EchoSink(httpd=httpd.Sink(control=control))
```

(continues on next page)

(continued from previous page)

```
def main():
    run(Component(call=echo_server, input=EchoSource),
        EchoDrivers(httpd=httpd.make_driver()))


if __name__ == '__main__':
    main()
```

CHAPTER 3

Indices and tables

- genindex
- modindex
- search