What is Japronto

Japronto is a screaming-fast, scalable, asynchronous Python 3.5+ HTTP toolkit integrated with pipelining HTTP server based on uvloop and picohttpparser.

Why to use Japronto

Imagine handling 1 million requests with python seems far fetched but Japronto has managed to achieve this feat. Japronto is designed for special tasks that could not be accomplished with bloated mainstream frameworks. It is a perfect fit for problems where every nanosecond counts.

How is Japronto different

Speed: Japronto is designed to handle upto 1 million requests per second, which is incredibly fast compared to other frameworks.
Lightweight: All core components of the framework are also implemented in C. A wide variety of low-level optimizations and tricks are used to tweak performance which make it lightweight.
Reliable: Despite being fast and lightweight, japronto is very reliable.

Installation

The installation of japronto is fairly simple and quick. It can be installed using Pip as below,

```
$ pip install japronto
```

Note: Japronto can only be installed on Linux based systems, because ‘uvloop’ one of japronto’s components is not yet supported for windows

Example

```python
import asyncio
from japronto import Application

# This is a synchronous handler.
def synchronous(request):
    return request.Response(text='I am synchronous!')

# This is an asynchronous handler, it spends most of the time in the event loop.
# It wakes up every second 1 to print and finally returns after 3 seconds.
# This does let other handlers to be executed in the same processes while
# from the point of view of the client it took 3 seconds to complete.
async def asynchronous(request):
    for i in range(1, 4):
        await asyncio.sleep(1)
        print(i, 'seconds elapsed')

    return request.Response(text='3 seconds elapsed')


app = Application()

r = app.router
r.add_route('/sync', synchronous)
r.add_route('/async', asynchronous)

app.run()
```

You can run the example program with the following command in the terminal

```
python japronto_demo.py
```

When you visit the following URLs on the browser respective outputs will be displayed

```
http://localhost:8080/sync
http://localhost:8080/async
```

Advantages

- Japronto is highly effective when it comes to speed for processing requests
- Simple routing
- Support for synchronous and asynchronous views
- Full support for HTTP pipelining

Disadvantages

- No being actively developed

Comparison with other Frameworks

What makes japronto different is HTTP pipelining which is crucial here since it’s one of the optimizations that Japronto takes into account when executing requests. In simple words, pipelining is a technique in which the client doesn’t need to wait for the response before sending subsequent requests over the same TCP connection. To ensure integrity of the communication, the server sends back several responses in the same order requests are received. Japronto is meant to provide a solid foundation for microservices using REST APIs with minimal overhead. In other words, there’s not much in the box. Developers only need to set up routing and decide which routes should use synchronous or asynchronous handlers.