
pycohttpparser Documentation

Release 1.0.0

Cory Benfield

March 21, 2015

1	Contents	3
1.1	pycohttpparser Python API	3
	Python Module Index	5

pycohttpparser is a Python wrapper library around the excellent [picohttpparser](#) C library. It aims to provide a speedy C implementation of the HTTP/1.1 parsing API used in the [hyper](#) project. Using CFFI, it supports both CPython 2.7 and 3.4, and PyPy.

To get started with pycohttpparser, simply install it from the cheeseshop:

```
$ pip install pycohttpparser
```

You'll need to make sure your system is set up for using CFFI. For more information, [consult CFFI's documentation](#).

Then, you can start parsing your HTTP messages! For example:

```
>>> import pycohttpparser.api as p
>>> message = socket.recv()
>>> m = memoryview(data)
>>> c = p.Parser()
>>> r = c.parse_request(m)
>>> r.method.tobytes()
b'POST'
>>> r.path.tobytes()
b'/post'
```


1.1 pycohttpparser Python API

This page documents pycohttpparser’s Python API.

An important feature to note is that, wherever possible, pycohttpparser uses `memoryview` objects to avoid copying data. The only objects that are not returned as “memoryview”s are response status codes and the HTTP minor version number.

class `pycohttpparser.api.Parser`

A single HTTP parser object. This object can parse HTTP requests and responses using `picohttpparser`.

This object is not thread-safe, and it does maintain state that is shared across parsing requests. For this reason, make sure that access to this object is synchronized if you use it across multiple threads.

parse_request (*buffer*)

Parses a single HTTP request from a buffer.

Parameters *buffer* – A `memoryview` object wrapping a buffer containing a HTTP request.

Returns A `Request` object, or `None` if there is not enough data in the buffer.

parse_response (*buffer*)

Parses a single HTTP response from a buffer.

Parameters *buffer* – A `memoryview` object wrapping a buffer containing a HTTP response.

Returns A `Response` object, or `None` if there is not enough data in the buffer.

class `pycohttpparser.api.Request`

`Request(method, path, minor_version, headers, consumed)`

consumed

Alias for field number 4

count (*value*) → integer – return number of occurrences of value

headers

Alias for field number 3

index (*value*[, *start*[, *stop*]]) → integer – return first index of value.

Raises `ValueError` if the value is not present.

method

Alias for field number 0

minor_version

Alias for field number 2

path

Alias for field number 1

class `pycohttpparser.api.Response`

`Response(status, msg, minor_version, headers, consumed)`

consumed

Alias for field number 4

count (*value*) → integer – return number of occurrences of value

headers

Alias for field number 3

index (*value*[, *start*[, *stop*]]) → integer – return first index of value.
Raises `ValueError` if the value is not present.

minor_version

Alias for field number 2

msg

Alias for field number 1

status

Alias for field number 0

class `pycohttpparser.api.ParseError`

An invalid HTTP message was passed to the parser.

h

`hyper`, 3

C

consumed (pycohttpparser.api.Request attribute), 3
consumed (pycohttpparser.api.Response attribute), 4
count() (pycohttpparser.api.Request method), 3
count() (pycohttpparser.api.Response method), 4

H

headers (pycohttpparser.api.Request attribute), 3
headers (pycohttpparser.api.Response attribute), 4
hyper (module), 3

I

index() (pycohttpparser.api.Request method), 3
index() (pycohttpparser.api.Response method), 4

M

method (pycohttpparser.api.Request attribute), 3
minor_version (pycohttpparser.api.Request attribute), 3
minor_version (pycohttpparser.api.Response attribute), 4
msg (pycohttpparser.api.Response attribute), 4

P

parse_request() (pycohttpparser.api.Parser method), 3
parse_response() (pycohttpparser.api.Parser method), 3
ParseError (class in pycohttpparser.api), 4
Parser (class in pycohttpparser.api), 3
path (pycohttpparser.api.Request attribute), 4

R

Request (class in pycohttpparser.api), 3
Response (class in pycohttpparser.api), 4

S

status (pycohttpparser.api.Response attribute), 4