
Pando Documentation

Release 0.47-dev

Chad Whitacre et al.

Jun 07, 2021

Contents

1	Installation	3
2	Contents	5
2.1	Tutorial	5
2.2	Reference	6
2.3	Deploying a Pando application	10
Python Module Index		13
Index		15

This is Pando, a Python web framework.

Pando's source code is on [GitHub](#), and is [MIT-licensed](#).

CHAPTER 1

Installation

pando is available on PyPI:

```
$ pip install pando
```


CHAPTER 2

Contents

2.1 Tutorial

2.1.1 Quick Start

Given: [POSIX](#) and [virtualenv](#)

Step 1: Make a sandbox:

```
$ virtualenv foo  
$ cd foo  
$ . bin/activate
```

Step 2: Install pando from PyPI:

```
(foo)$ pip install pando  
blah  
blah  
blah
```

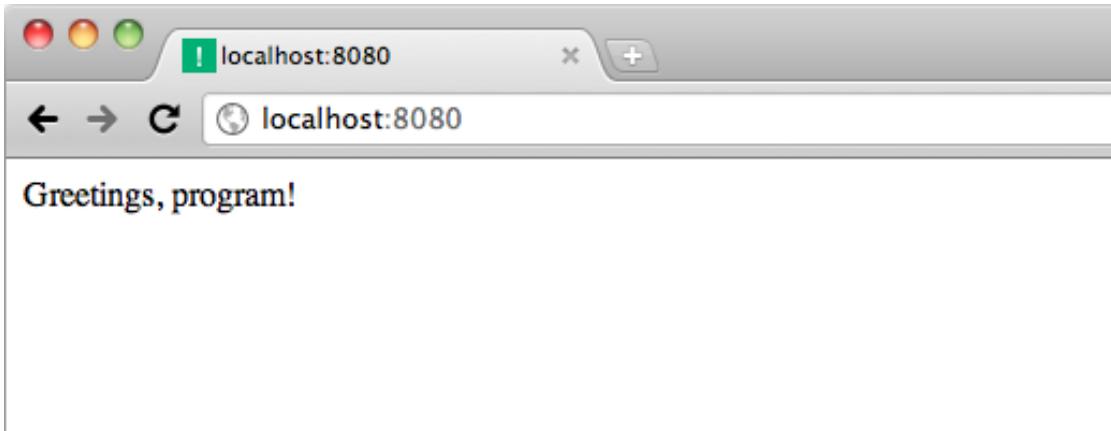
Step 3: Create a website root:

```
(foo)$ mkdir www  
(foo)$ cd www
```

Step 4: Create a web page, and start pando inside it:

```
(foo)$ echo Greetings, program! > index.html.spt  
(foo)$ python -m pando  
Greetings, program! Welcome to port 8080.
```

Step 5: Check localhost for your new page!



2.2 Reference

This is the API reference for the Pando library.

2.2.1 body_parsers

This module contains Pando's built-in body parsers.

Body parsers are optional ways to enable Pando to uniformly parse POST body content according to its supplied Content-Type.

A body parser has the signature:

```
def name(raw, headers):
```

where `raw` is the raw bytestring to be parsed, and `headers` is the Headers mapping of the supplied headers.

```
pando.body_parsers.formdata(raw, headers)
```

Parse `raw` as form data.

Supports application/x-www-form-urlencoded and multipart/form-data.

```
pando.body_parsers.jsondata(raw, headers)
```

Parse `raw` as JSON data.

2.2.2 exceptions

Custom exceptions raised by Pando

```
exception pando.exceptions.CRLFInjection
```

A 400 `Response` (per #249) raised if there's a suspected CRLF Injection attack in the headers.

```
exception pando.exceptions.MalformedBody(msg)
```

A 400 `Response` raised if parsing the body of a POST request fails.

```
exception pando.exceptions.UnknownBodyType(ctype)
```

A 415 `Response` raised if the Content-Type of the body of a POST request doesn't have a `body_parser` registered for it.

```
exception pando.exceptions.BadLocation(msg)
```

A 500 `Response` raised if an invalid redirect is attempted.

2.2.3 http

baseheaders

```
class pando.http.baseheaders.BaseHeaders (headers=())
Bases: pando.http.mapping.BytesMapping, pando.http.mapping.
CaseInsensitiveMapping
```

Represent the headers in an HTTP Request or Response message.

How to send non-English unicode string using HTTP header? and What character encoding should I use for a HTTP header? have good notes on why we do everything as pure bytes here.

__init__ (*headers*=())

Takes headers as a dict, or list of items.

__setitem__ (*name*, *value*)

Checks for CRLF in value, then calls the superclass method:

CaseInsensitiveMapping.**__setitem__** (*name*, *value*)

add (*name*, *value*)

Checks for CRLF in value, then calls the superclass method:

CaseInsensitiveMapping.**add** (*name*, *value*)

raw

Return the headers as a bytestring, formatted for an HTTP message.

__contains__ (*k*) → True if D has a key k, else False

__getitem__ (*name*)

__slots__ = ('encoding', 'encoding_errors')

all (*name*)

encoding

encoding_errors

get (*name*, *default=None*)

keyerror (*name*)

Raises a 400 *Response*.

ones (**names*)

Given one or more names of keys, return a list of their values.

pop (*name*, *default=<object object>*)

popall (*name*, **default*)

D.pop(k[d]) -> v, remove specified key and return the corresponding value. If key is not found, d is returned if given, otherwise KeyError is raised

mapping

```
class pando.http.mapping.Mapping (*a, **kw)
Bases: aspen.http.mapping.Mapping
```

__init__ (**a*, ***kw*)

Initializes the mapping.

Loops through positional arguments first, then through keyword args.

Positional arguments can be dicts or lists of items.

keyerror (*name*)

Raises a 400 *Response*.

__getitem__ (*name*)

Given a name, return the last value or call self.keyerror.

__setitem__ (*name, value*)

Given a name and value, clobber any existing values.

add (*name, value*)

Given a name and value, clobber any existing values with the new one.

all (*name*)

Given a name, return a list of values, possibly empty.

get (*name, default=None*)

Override to only return the last value.

ones (**names*)

Given one or more names of keys, return a list of their values.

pop (*name, default=<object object>*)

Given a name, return a value.

This removes the last value from the list for name and returns it. If there was only one value in the list then the key is removed from the mapping. If name is not present and default is given, that is returned instead. Otherwise, self.keyerror is called.

popall ()

D.pop(k[d]) -> v, remove specified key and return the corresponding value. If key is not found, d is returned if given, otherwise KeyError is raised

class pando.http.mapping.CaseInsensitiveMapping (**a*, ***kw*)

Bases: *pando.http.mapping.Mapping*

__contains__ (*k*) → True if D has a key k, else False

__getitem__ (*name*)

__setitem__ (*name, value*)

add (*name, value*)

get (*name, default=None*)

all (*name*)

pop (*name, default=<object object>*)

popall (*name*)

D.pop(k[d]) -> v, remove specified key and return the corresponding value. If key is not found, d is returned if given, otherwise KeyError is raised

__init__ (**a*, ***kw*)

Initializes the mapping.

Loops through positional arguments first, then through keyword args.

Positional arguments can be dicts or lists of items.

keyerror (*name*)

Raises a 400 *Response*.

ones(*names)

Given one or more names of keys, return a list of their values.

class pando.http.mapping.**BytesMapping**(*a, **kw)

Bases: *pando.http.mapping.Mapping*

This mapping automatically transcodes keys and values.

Attributes: encoding (str): UTF-8 by default encoding_errors (str): ‘backslashreplace’ by default

```
>>> m = BytesMapping()
>>> m[b'foo'] = b'bar'
>>> m[b'foo']
b'bar'
>>> m['foo']
'bar'
```

__slots__ = ('encoding', 'encoding_errors')

encoding

encoding_errors

__contains__(k) → True if D has a key k, else False

__getitem__(name)

__setitem__(name, value)

add(name, value)

get(name, default=None)

all(name)

pop(name, default=<object object>)

popall(name, *default)

D.pop(k,d) -> v, remove specified key and return the corresponding value. If key is not found, d is returned if given, otherwise KeyError is raised

keyerror(name)

Raises a 400 *Response*.

ones(*names)

Given one or more names of keys, return a list of their values.

response**class** pando.http.response.**CloseWrapper**(request, body)

Conform to WSGI’s facility for running code *after* a response is sent.

__iter__()

close()

exception pando.http.response.**Response**(code=200, body='', headers=None)

Represent an HTTP Response message.

request = None

whence_raised = (None, None)

`__init__(code=200, body='', headers=None)`

Takes an int, a string, a dict.

- code an HTTP response code, e.g., 404
- body the message body as a string
- headers a dict, list, or bytestring of HTTP headers

Code is first because when you're raising your own Responses, they're usually error conditions. Body is second because one more often wants to specify a body without headers, than a header without a body.

`to_wsgi(environ, start_response, charset)`

`__repr__() <==> repr(x)`

`__str__() <==> str(x)`

`set_whence_raised()`

Sets and returns the value of `self.whence_raised`.

It's a tuple, (filename, linenum) where we were raised from.

This function needs to be called from inside the `except` block.

2.2.4 logging

Pando logging convenience wrappers

`pando.logging.log(*messages, **kw)`

Make logging more convenient - use magic to get the `__name__` of the calling module/function and log as it.

'level' if present as a kwarg, is the level to log at. 'upframes' if present as a kwarg, is how many frames up to look for the name.

other kwargs are passed through to `Logger.log()`

`pando.logging.log_dammit(*messages, **kw)`

like `log()`, but critical instead of warning

2.2.5 testing

2.3 Deploying a Pando application

Pando applications are standard WSGI applications and should work with any WSGI server, for example [Gunicorn](#).

2.3.1 Client IP addresses

If your application uses the `pando.http.request.Request.source` property, then you need to ensure that the `trusted_proxies` attribute of the `website` object is correctly set. Here are a few examples:

- If you don't use any reverse proxy, then `trusted_proxies` should be left empty.
- If you use [cloudflare](#), then `trusted_proxies` should also be left empty.
- If you only use local-network load balancers, then you can set `trusted_proxies` to `[['private']]`.
- If you use proxies at specific IP addresses, then `trusted_proxies` should contain those addresses, e.g. `[['x.x.x.x', 'y.y.y.y']]`.

- If you use both of the above, then *trusted_proxies* should contain the two lists, i.e. `[['private'], ['x.x.x', 'y.y.y.y']]`.

Python Module Index

p

`pando`, 6
`pando.body_parsers`, 6
`pando.exceptions`, 6
`pando.http`, 6
`pando.http.baseheaders`, 7
`pando.http.mapping`, 7
`pando.http.response`, 9
`pando.logging`, 10
`pando.testing`, 10

Symbols

—contains__() (pando.http.baseheaders.BaseHeaders method), 7
—contains__() (pando.http.mapping.BytesMapping method), 9
—contains__() (pando.http.mapping.CaseInsensitiveMapping method), 8
—getitem__() (pando.http.baseheaders.BaseHeaders method), 7
—getitem__() (pando.http.mapping.BytesMapping method), 9
—getitem__() (pando.http.mapping.CaseInsensitiveMapping method), 8
—getitem__() (pando.http.mapping.Mapping method), 8
—init__() (pando.http.baseheaders.BaseHeaders method), 7
—init__() (pando.http.mapping.CaseInsensitiveMapping method), 8
—init__() (pando.http.mapping.Mapping method), 7
—init__() (pando.http.response.Response method), 9
—iter__() (pando.http.response.CloseWrapper method), 9
—repr__() (pando.http.response.Response method), 10
—setitem__() (pando.http.baseheaders.BaseHeaders method), 7
—setitem__() (pando.http.mapping.BaseHeaders.CaseInsensitiveMapping method), 7
—setitem__() (pando.http.mapping.BytesMapping method), 9
—setitem__() (pando.http.mapping.CaseInsensitiveMapping method), 8
—setitem__() (pando.http.mapping.Mapping method), 8
—slots__ (pando.http.baseheaders.BaseHeaders attribute), 7
—slots__ (pando.http.mapping.BytesMapping attribute), 7
tribute), 9
str__() (pando.http.response.Response method), 10
A
add() (pando.http.baseheaders.BaseHeaders method), 7
add() (pando.http.mapping.BaseHeaders.CaseInsensitiveMapping method), 7
add() (pando.http.mapping.BytesMapping method), 9
add() (pando.http.mapping.CaseInsensitiveMapping method), 8
add() (pando.http.mapping.Mapping method), 8
all() (pando.http.baseheaders.BaseHeaders method), 7
all() (pando.http.mapping.BytesMapping method), 9
all() (pando.http.mapping.CaseInsensitiveMapping method), 8
all() (pando.http.mapping.Mapping method), 8
BadLocation, 6
BaseHeaders (class in pando.http.baseheaders), 7
BytesMapping (class in pando.http.mapping), 9
B
CaseInsensitiveMapping (class in pando.http.mapping), 8
close() (pando.http.response.CloseWrapper method), 9
CloseWrapper (class in pando.http.response), 9
CRLFInjection, 6
encoding (pando.http.baseheaders.BaseHeaders attribute), 7
encoding (pando.http.mapping.BytesMapping attribute), 9
encoding_errors (pando.http.baseheaders.BaseHeaders attribute), 7

encoding_errors (*pando.http.mapping.BytesMapping*.pop () (*pando.http.baseheaders.BaseHeaders* method),
attribute), 9

F

formdata () (*in module pando.body_parsers*), 6

G

get () (*pando.http.baseheaders.BaseHeaders* method),
7

get () (*pando.http.mapping.BytesMapping* method), 9
get () (*pando.http.mapping.CaseInsensitiveMapping* method), 8

get () (*pando.http.mapping.Mapping* method), 8

J

jsondata () (*in module pando.body_parsers*), 6

K

keyerror () (*pando.http.baseheaders.BaseHeaders* method), 7

keyerror () (*pando.http.mapping.BytesMapping* method), 9

keyerror () (*pando.http.mapping.CaseInsensitiveMapping* method), 8

keyerror () (*pando.http.mapping.Mapping* method), 8

L

log () (*in module pando.logging*), 10

log_dammit () (*in module pando.logging*), 10

M

MalformedBody, 6

Mapping (*class in pando.http.mapping*), 7

O

ones () (*pando.http.baseheaders.BaseHeaders* method),
7

ones () (*pando.http.mapping.BytesMapping* method), 9

ones () (*pando.http.mapping.CaseInsensitiveMapping* method), 8

ones () (*pando.http.mapping.Mapping* method), 8

P

pando (*module*), 6

pando.body_parsers (*module*), 6

pando.exceptions (*module*), 6

pando.http (*module*), 6

pando.http.baseheaders (*module*), 7

pando.http.mapping (*module*), 7

pando.http.response (*module*), 9

pando.logging (*module*), 10

pando.testing (*module*), 10

7
pop () (*pando.http.mapping.BytesMapping* method), 9

pop () (*pando.http.mapping.CaseInsensitiveMapping* method), 8
pop () (*pando.http.mapping.Mapping* method), 8

popall () (*pando.http.baseheaders.BaseHeaders* method), 7
popall () (*pando.http.mapping.BytesMapping* method), 9

popall () (*pando.http.mapping.CaseInsensitiveMapping* method), 8
popall () (*pando.http.mapping.Mapping* method), 8

R

raw (*pando.http.baseheaders.BaseHeaders* attribute), 7
request (*pando.http.response.Response* attribute), 9
Response, 9

S

set_whence_raised ()
(*pando.http.response.Response* method),
10

T

to_wsgi () (*pando.http.response.Response* method), 10

U

UnknownBodyType, 6

W

whence_raised (*pando.http.response.Response* attribute), 9