
strawpoll.py Documentation

Release 0.2.0

PapyrusThePlant

Feb 02, 2018

Contents

1	A note on coroutines	3
2	API reference	5
2.1	Version info	5
2.2	API	5
2.3	Poll	6
2.4	Exceptions	7
3	Indices and tables	9
	Python Module Index	11

Contents:

CHAPTER 1

A note on coroutines

A coroutine is a function that is either decorated with `@asyncio.coroutine` for python 3.4 or defined with `async def` in python 3.5+.

Coroutines are not like regular functions, if you invoke it as usual, it will not be executed. The proper way to call a coroutine is to use `yield from` in python 3.4 or `await` in python 3.5+, like so:

```
# With python 3.4
@asyncio.coroutine
def my_coro():
    yield from another_coro()

# With python 3.5+
async def my_coro():
    await another_coro()
```

As you can see async programming has slight syntax nuances between python 3.4 and python 3.5+. **It is highly recommended to use python 3.5+.**

An async python wrapper for the Strawpoll API.

2.1 Version info

`strawpoll.version_info`

A named tuple containing the five components of the version number in a similar manner as `sys.version_info`.

`strawpoll.__version__`

A string representation of `version_info`.

2.2 API

class `strawpoll.API` (*, `loop=None`, `requests_policy=<RequestsPolicy.asynchronous: 0>`)

Represents the strawpoll API.

Parameters `loop` – The event loop to use for the asynchronous calls. Defaults to `None`, which will grab the default event loop instead.

get_poll (`arg`, *, `request_policy=None`)

Retrieves a poll from strawpoll.

Parameters

- `arg` – Either the ID of the poll or its strawpoll url.
- `request_policy` (Optional[`RequestsPolicy`]) – Overrides `API.requests_policy` for that request.

Raises `HTTPException` – Requesting the poll failed.

Returns A poll constructed with the requested data.

Return type `Poll`

submit_poll (*poll*, *, *request_policy=None*)

Submits a poll on strawpoll.

Parameters

- **poll** (*Poll*) – The poll to submit.
- **request_policy** (Optional[*RequestsPolicy*]) – Overrides `API.requests_policy` for that request.

Raises

- *ExistingPoll* – This poll instance has already been submitted.
- *HTTPException* – The submission failed.

Returns The given poll updated with the data sent back from the submission.

Return type *Poll*

Note: Only polls that have a non empty title and between 2 and 30 options can be submitted.

class `strawpoll.RequestsPolicy`

An enumeration of the requests' policies.

asynchronous = 0

The asynchronous policy.

synchronous = 1

The synchronous policy.

2.3 Poll

class `strawpoll.Poll` (*title*, *options*, ***kwargs*)

Represents a poll from strawpoll.

Parameters

- **title** (*str*) – The question that the poll is asking.
- **options** (*iterable*) – An iterable of elements accepting the `str()` operation.

Variables

- **id** (*int*) – The poll ID on strawpoll. If the poll has not been submitted on or retrieved from strawpoll, it's None instead.
- **multi** (*bool*) – Specifies if the polls accepts multiple votes from one user. Defaults to False.
- **dupcheck** (*str*) – Defines how to handle checking for duplicate votes. Defaults to normal.
- **captcha** (*bool*) – Specifies if the poll requires users to pass a captcha to vote. Defaults to False.

options

Returns a list of strings to represent the options for the poll, in the order they were given when the poll was created.

votes

Return a list of integers that correspond to the same indexed option which specify the current votes for that option.

total_votes

Returns the total number of votes on the poll.

url

Returns the url of the poll. If the poll has not been submitted yet, an empty string is returned instead.

results (*limit=None*)

Returns a list of tuples each containing a string representing an option and an int which specify the current votes for that option, ordered by their votes count.

Parameters **limit** (*int*) – The maximum number of results to return. If not specified, every results will be returned.

result_at (*index*)

Returns a tuple containing a string representing the option and an int which specify the current votes for that option.

Parameters **index** (*int*) – The index of the wanted option in the options list.

2.4 Exceptions

exception `strawpoll.StrawpollException`

Base class of all the other exceptions in strawpoll.py

exception `strawpoll.ExistingPoll`

Exception thrown when an attempt to submit a poll that already has been submitted is made.

exception `strawpoll.HTTPException` (*response, data*)

Exception thrown when an HTTP request failed.

Variables

- **response** (`aiohttp.ClientResponse`) – The failed HTTP request's response.
- **text** (*str*) – The error message extracted from the response. Can be an empty string.
- **code** (*int*) – The error code extracted from the response. Can be 0 if no error code was found.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

S

strawpoll, 3

Symbols

`__version__` (in module `strawpoll`), 5

A

`API` (class in `strawpoll`), 5

`asynchronous` (`strawpoll.RequestsPolicy` attribute), 6

E

`ExistingPoll`, 7

G

`get_poll()` (`strawpoll.API` method), 5

H

`HTTPException`, 7

O

`options` (`strawpoll.Poll` attribute), 6

P

`Poll` (class in `strawpoll`), 6

R

`RequestsPolicy` (class in `strawpoll`), 6

`result_at()` (`strawpoll.Poll` method), 7

`results()` (`strawpoll.Poll` method), 7

S

`strawpoll` (module), 3

`StrawpollException`, 7

`submit_poll()` (`strawpoll.API` method), 5

`synchronous` (`strawpoll.RequestsPolicy` attribute), 6

T

`total_votes` (`strawpoll.Poll` attribute), 7

U

`url` (`strawpoll.Poll` attribute), 7

V

`version_info` (in module `strawpoll`), 5

`votes` (`strawpoll.Poll` attribute), 6