
snowman Documentation

Release 0.1.0

rui oaix

Jul 09, 2017

Contents:

1	Tool: snowman	1
2	Developer Interface	3
2.1	Info	3
2.2	Profit	4
2.3	Analysis	4
2.4	History	6
3	Indices and tables	9

CHAPTER 1

Tool: snowman

usage:

```
>>> snowman verb object [-h] [-v] [-D] [-O] [-f OFILE] [-P PARAMS [PARAMS ...]]
```

positional arguments: verb specify the action snowman need to perform. object the object related to the action.

available verb:

```
$ snowman info ZHXXXXXX
$ snowman owner ZHXXXXXX
$ snowman holdings ZHXXXXXX
$ snowman profit ZHXXXXXX
$ snowman benefit ZHXXXXXX
$ snowman maxdraw ZHXXXXXX
$ snowman turnover ZHXXXXXX
$ snowman liquidity ZHXXXXXX
$ snowman volatility ZHXXXXXX
$ snowman topstocks ZHXXXXXX
$ snowman analysis ZHXXXXXX
$ snowman history ZHXXXXXX
```

optional arguments:

- | | |
|-----------------------|--|
| -h, --help | show this help message and exit |
| -v, --version | print the snowman version and exit. |
| -D, --debug | show all messages, including debug messages. |
| -O, --origin | return the origin content from xueqiu instead of the simple version. |
| -f OFILE | save the result data into specified file. |
| -P params-list | parameters for actions. |

Example:

```
$ snowman info ZH123456 #return the basic information of portfolio ZH123456 in simple_
↪format.
{'symbol': 'ZH123456', 'name': '', 'market': 'cn', 'status': 'active', 'created':
↪'2015.01.07', 'updated_at': '2017-05-11 04:04:13', 'net_value': 1.3235, 'follower_
↪count': 1}
$ snowman info ZH123456 -O # return the basic information of portfolio ZH123456 in_
↪origin format.
$ snowman profit ZH123456 -P 33 # return the latest 33 days profit data of portfolio_
↪ZH123456 in simple format.
$ snowman topstocks ZH123456 -P 20 # return the top 20 best-profit stocks of porfolio_
↪ZH123456
$ snowman history ZH123456 -P 30 2 10 # return 30 rebalancing records starting from_
↪10.
```

Info

class `snowman.Info` (*symbol, session=None*)
the basic information of a portfolio.

Parameters

- **symbol** – the portfolio symbol, a string starting with 'ZH'
- **session** – (optional) the session object responsible for http(s) connection

Usage:

```
>>> from snowman import Info
>>> info = Info('ZH123456')
>>> info.get()
>>> info.owner()
>>> info.holdings()
```

get (*origin=False, update=False*)
request the data when necessary, cache the result and return it.

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict

holdings (*origin=False*)
return the current holdings of the portfolio, which is contained as a part of the basic information.

Parameters **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.

Return type dict

owner (*origin=False*)

return the owner information which is contained as a part of the basic information.

Parameters **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.

Return type dict

Profit

class snowman.**Profit** (*symbol, session=None*)

the profit history of a portfolio.

Parameters

- **symbol** – the portfolio symbol, a string starting with 'ZH'
- **session** – (optional) the session object responsible for http(s) connection

Usage:

```
>>> from snowman import Profit
>>> pft = Profit('ZH123456')
>>> pft.get() # return all the profit history
>>> pft.get(10) # return the profit history in the latest 10 days.
```

get (*num=0, origin=False, update=False*)

request the profit history data when necessary, cache the data and return it.

Parameters

- **num** – num param only works when origin is False, can't be negative. When num is 0, return the history data starting from the creation date of the portfolio, when num is postive integer N, only return the history data starting from N num ago.
- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type list

Analysis

class snowman.**Analysis** (*symbol, session=None*)

the analysis information of a portfolio.

Parameters

- **symbol** – the portfolio symbol, a string starting with 'ZH'
- **session** – (optional) the session object responsible for http(s) connection

Usage:


```
>>> from snowman import Analysis
>>> ana = Analysis('ZH123456')
>>> ana.benefit() # return benefit analysis data
>>> ana.turnover() # return turnover analysis data
>>> ...
```

all (*page=1, count=5, origin=False*)
update all the caches, and return all analysis data in a single dict.

Parameters

- **page** – (optional, defaults to 1) this only affect the data for top stocks. this param works with count param. if page is N, the result is stocks ranking between $[(N - 1) * \text{count} + 1, N * \text{count}]$ so when page is 1, the result is top [count] stocks.
- **count** – (optional, defaults to 5) this only affect the data for top stocks. Check page param.
- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict

benefit (*origin=False, update=False*)
benefit analysis data

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type list

liquidity (*origin=False, update=False*)
liquidity analysis data

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict if origin is True, otherwise an float number

max_draw (*origin=False, update=False*)
max draw analysis data

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict

top_stocks (*page=1, count=5, origin=False, update=False*)
volatility analysis data

Parameters

- **page** – (optional, defaults to 1) this param works with count param. if page is N, the result is stocks ranking between $[(N - 1) * \text{count} + 1, N * \text{count}]$ so when page is 1, the result is top [count] stocks.
- **count** – (optional, defaults to 5) check page param.
- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict if origin is True, otherwise list

turnover (*origin=False, update=False*)
turnover analysis data

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict if origin is True, otherwise an float number

volatility (*origin=False, update=False*)
volatility analysis data

Parameters

- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customized version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict if origin is True, otherwise an float number

History

class snowman.**History** (*symbol, session=None*)
the rebalancing history information of a portfolio.

Parameters

- **symbol** – the portfolio symbol, a string starting with 'ZH'
- **session** – (optional) the session object responsible for http(s) connection

Usage:

```
>>> from snowman import History
>>> his = History('ZH123456')
>>> his.get() #query all the rebalancing records of portfolio ZH123456.
>>> his.get(10) #query the latest 10 records of portfolio ZH123456
>>> his.get(10, 1, 3)
```

```
>>> # query the latest 10 records, if there are no less than 12 records,
>>> # 4 requests will be sent,the count of result will be 12.
>>> # otherwise, just return all the records.
>>> his.get(origin = True) # return all the records in origin version.
>>> his.get(page = 1, count = 3) #return all the records, query 3 records each_
↪time.
```

be_login()

make sure snowman has logged xueqiu.com.

get (*num=0, page=1, count=20, origin=False, update=False*)

request the history data when necessary, cache the data and return it.

Parameters

- **num** – (optional, defaults to 0) this param specifies how many rebalance records snowman needs to query at least. The actual count of the result depends on history, page and count. these three works together to decide the result. when num is 0, all the records will be queried.
- **page** – (optional, defaults to 1) this param works with count param, each request will return all the records in the range [(page - 1) * count + 1, page * count]. When num is positive number N, snowman will send mutiple requests, each request return count records. After the last request, the total number of records is (page * count), which will be not smaller than num or there are no more records.
- **count** – (optional, defaults to 20) check param page.
- **origin** – (optional, defaults to False) return the origin data from the api when origin is true, otherwise, return a simple customed version.
- **update** – (optional, defaults to False) try to reuse the cache if update is false, otherwise, request and update the cache.

Return type dict

CHAPTER 3

Indices and tables

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

A

`all()` (snowman.Analysis method), 5
`Analysis` (class in snowman), 4

B

`be_login()` (snowman.History method), 7
`benefit()` (snowman.Analysis method), 5

G

`get()` (snowman.History method), 7
`get()` (snowman.Info method), 3
`get()` (snowman.Profit method), 4

H

`History` (class in snowman), 6
`holdings()` (snowman.Info method), 3

I

`Info` (class in snowman), 3

L

`liquidity()` (snowman.Analysis method), 5

M

`max_draw()` (snowman.Analysis method), 5

O

`owner()` (snowman.Info method), 4

P

`Profit` (class in snowman), 4

T

`top_stocks()` (snowman.Analysis method), 5
`turnover()` (snowman.Analysis method), 6

V

`volatility()` (snowman.Analysis method), 6