

银河启睿策略中心

Search docs

参数配置

- [参数配置](#)

API 手册

- [入口函数](#)
- [约定函数](#)
- [交易接口](#)
- [仓位查询接口](#)
- [数据查询接口](#)
 - [市场基础数据](#)
 - [all_instruments](#) - 所有合约基础信息
 - [instruments](#) - 合约详细信息
 - [get_trading_dates](#) - 交易日列表
 - [get_previous_trading_date](#) - 上一交易日
 - [get_next_trading_date](#) - 下一交易日
 - [get_yield_curve](#) - 收益率曲线
 - [行情](#)
 - [history_bars](#) - 某一合约历史 bar 数据
 - [current_snapshot](#) - 当前快照数据
 - [history_ticks](#) - 指定合约的历史 tick 数据
 - [get_price](#) - 合约历史数据
 - [get_price_change_rate](#) - 历史涨跌幅
 - [股票](#)
 - [industry](#) - 行业股票列表
 - [sector](#) - 板块股票列表
 - [get_dividend](#) - 获取分红数据
 - [is_suspended](#) - 全天停牌判断
 - [is_st_stock](#) - ST股判断
 - [get_split](#) - 拆分数据
 - [get_securities_margin](#) - 融资融券信息
 - [concept](#) - 概念股列表
 - [get_margin_stocks](#) - 融资融券列表
 - [get_shares](#) - 流通股信息
 - [get_turnover_rate](#) - 历史换手率
 - [get_factor](#) - 因子
 - [get_industry](#) - 行业股票列表
 - [get_instrument_industry](#) - 股票行业分类
 - [get_stock_connect](#) - 沪深港通持股信息
 - [current_performance](#) - 财务快报数据
 - [get_pit_financials_ex](#) - 季度财务信息
 - [指数](#)
 - [index_components](#) - 指数成分股
 - [index_weights](#) - 指数成分股权重
 - [宏观经济](#)
 - [econ.get_reserve_ratio](#) - 存款准备金率
- [其他接口](#)
- [scheduler 定时器](#)
- [类](#)
- [枚举常量](#)

银河启睿策略中心

- »
- 数据查询接口
- [View page source](#)

数据查询接口

市场基础数据

all_instruments - 所有合约基础信息

deepquant. quest. api. all_instruments(*type=None, date=None*)

获取某个国家市场的所有合约信息。使用者可以通过这一方法很快地对合约信息有一个快速了解，目前仅支持中国市场。

Parameters:

- type** (Optional[str]) -- 需要查询合约类型，例如：type='stock'代表股票。默认是所有类型
- date** (Union[str, datetime, date, None]) -- 查询时间点

其中type参数传入的合约类型和对应的解释如下：

合约类型	说明
stock	Common Stock, 即股票
ETF	Exchange Traded Fund, 即交易所交易基金
LOF	Listed Open-Ended Fund，即上市型开放式基金
index	Index, 即指数
future	Futures，即期货，包含股指、国债和商品期货

Return type:

DataFrame

instruments - 合约详细信息

deepquant. quest. api. instruments(*id_or_symbols*)

获取某个国家市场内一个或多个合约的详细信息。目前仅支持中国市场。

Parameters:

id_or_symbols (Union[str, List[str]]) -- 合约代码或者合约代码列表

Example:

- 获取单一股票合约的详细信息:

```
1 [In]instruments('000001.SZ')
2 [Out]
3 Instrument(market_code=000001.SZ, symbol=平安银行, abbrev_symbol=PAYH, listed_date=19910403, de_listed_date=null, board_type=MainBoard, sector_code_name=金融, sector_code=Financials, round_lot=100, exchange=SZ,
```

- 获取多个股票合约的详细信息:

```
1 [In]instruments(['000001.SZ', '000024.SZ'])
2 [Out]
3 [Instrument(market_code=000001.SZ, symbol=平安银行, abbrev_symbol=PAYH, listed_date=19910403, de_listed_date=null, board_type=MainBoard, sector_code_name=金融, sector_code=Financials, round_lot=100, exchange=SZ
```

- 获取合约已上市天数:

```
1 instruments('000001.SZ').days_from_listed()
```

- 获取合约距离到期天数:

```
1 instruments('IF1701').days_to_expire()
```

Return type:

```
Union[Instrument, List[Instrument]]
```

get_trading_dates - 交易日列表[🔗](#)

deepquant.quest.api.get_trading_dates(*start_date, end_date*)[🔗](#)

获取某个国家市场的交易日列表（起止日期加入判断）。目前仅支持中国市场。

Parameters:

- start_date** (Union[str, date, datetime, Timestamp]) -- 开始日期
- end_date** (Union[str, date, datetime, Timestamp]) -- 结束如期

Return type:

```
DatetimeIndex
```

get_previous_trading_date - 上一交易日[🔗](#)

deepquant.quest.api.get_previous_trading_date(*date*)[🔗](#)

获取指定日期的之前的第 n 个交易日。

Parameters:

- date** (Union[str, date, datetime, Timestamp]) -- 指定日期
- n** (Optional[int]) --

Example:

```
1 [In]get_previous_trading_date(date='2016-05-02')
2 [Out]
3 [datetime.date(2016, 4, 29)]
```

Return type:

```
date
```

get_next_trading_date - 下一交易日[🔗](#)

deepquant.quest.api.get_next_trading_date(*date*)[🔗](#)

获取指定日期之后的第 n 个交易日

Parameters:

- date** (Union[str, date, datetime, Timestamp]) -- 指定日期
- n** (Optional[int]) --

Example:

```
1 [In]get_next_trading_date(date='2016-05-01')
2 [Out]
3 [datetime.date(2016, 5, 3)]
```

Return type:

```
date
```

get_yield_curve - 收益率曲线[🔗](#)

deepquant.quest.api.get_yield_curve(*date=None, tenor=None*)[🔗](#)

获取某个国家市场指定日期的收益率曲线水平。

数据为2002年至今的中债国债收益率曲线，来源于中央国债登记结算有限责任公司。

Parameters:

- date** (Union[str, date, datetime, Timestamp, None]) -- 查询日期，默认为策略当前日期前一天
- tenor** (Optional[str]) -- 标准期限，'0S' - 隔夜，'1M' - 1个月，'1Y' - 1年，默认为全部期限

Example:

```
1 [In]
2 get_yield_curve('20130104')
3
4 [Out]
5      0S      1M      2M      3M      6M      9M      1Y      2Y      2013-01-04  0.0196  0.0253  0.0288  0.0279  0.0280  0.0283  0.0292  0.0310
6
7      3Y      4Y      ...      6Y      7Y      8Y      9Y      10Y      2013-01-04  0.0314  0.0318  ...      0.0342  0.0350  0.0353  0.0357  0.0361
8 ...
```

Return type:

DataFrame

行情

history_bars - 某一合约历史 bar 数据

deepquant. quest. api. history_bars(*market_code, bar_count, frequency, fields=None, skip_suspended=True, include_now=False, adjust_type='pre'*)

获取指定合约的历史 k 线行情，同时支持日以及分钟历史数据。不能在init中调用。

日回测获取分钟历史数据：不支持

日回测获取日历史数据

调用时间	返回数据
T日before_trading	T-1日day bar

T日handle_bar T日day bar

分钟回测获取日历史数据

调用时间	返回数据
T日before_trading	T-1日day bar

T日handle_bar T-1日day bar

分钟回测获取分钟历史数据

调用时间	返回数据
T日before_trading	T-1日最后一个minute bar

T日handle_bar T日当前minute bar

Parameters:

- **market_code** (str) -- 合约代码
- **bar_count** (int) -- 获取的历史数据数量，必填项
- **frequency** (str) -- 获取数据什么样的频率进行。'1d'、'1m' 和 '1w' 分别表示每日、每分钟和每周，必填项
- **fields** (Union[str, List[str], None]) -- 返回数据字段。必填项。见下方列表。
- **skip_suspended** (Optional[bool]) -- 是否跳过停牌数据
- **include_now** (Optional[bool]) -- 是否包含当前数据
- **adjust_type** (Optional[str]) -- 复权类型，默认为前复权 pre；可选 pre, none, post

fields	字段名
datetime	时间戳
open_price	开盘价
high_price	最高价
low_price	最低价
close_price	收盘价
volume	成交量
total_turnover	成交额
open_interest	持仓量（期货专用）
basis_spread	期现差（股指期货专用）
settlement	结算价（期货日线专用）
prev_settlement	结算价（期货日线专用）

Example:

获取最近5天的日线收盘价序列（策略当前日期为20160706）：

```
1 [In]
2 logger.info(history_bars('000002.SZ', 5, '1d', 'close_price'))
3 [Out]
4 [ 8.69  8.7   8.71  8.81  8.81]
```

Return type:

ndarray

current_snapshot - 当前快照数据

deepquant. quest. api. current_snapshot(*market_code*)

获得当前市场快照数据。只能在日内交易阶段调用，获取当日调用时点的市场快照数据。 市场快照数据记录了每日从开盘到当前的数据信息，可以理解为一个动态的day bar数据。 在目前分钟回测中，快照数据为当日所有分钟线累积而成，一般情况下，最后一个分钟线获取到的快照数据应当与当日的日线行情保持一致。 需要注意，在实盘模拟中，该函数返回的是调用当时的市场快照情况，所以在同一个handle_bar中不同时间点调用可能返回的数据不同。 如果当日截止到调用时候对应股票没有任何成交，那么snapshot中的close, high, low, last几个价格水平都将以0表示。

Parameters:

d_or_symbol -- 合约代码或简称

Example:

在handle_bar中调用该函数，假设策略当前时间是20160104 09:33:

```
1 [In]
2 logger.info(current_snapshot('000001.SZ'))
3 [Out]
4 2016-01-04 09:33:00.00 INFO
5 Snapshot(market_code: '000001.SZ', datetime: datetime.datetime(2016, 1, 4, 9, 33), open: 10.0, high: 10.025, low: 9.9667, last: 9.9917, volume: 2050320, total_turnover: 20485195, prev_close: 9.99)
```

Return type:

Optional[[TickObject](#)]

history_ticks - 指定合约的历史 tick 数据

deepquant. quest. api. history_ticks(*market_code, count*)

获取指定合约历史（不晚于当前时间的）tick 对象，仅支持在 tick 级别的策略（回测、模拟交易、实盘）中调用。

Parameters:

- **market_code** (str) -- 合约代码
- **count** (int) -- 获取的 tick 数量

Return type:

List[[TickObject](#)]

get_price - 合约历史数据

deepquant. quest. api. get_price(*market_code, start_date, end_date=None, frequency='1d', fields=None, adjust_type='pre', skip_suspended=False, variety='stock', expect_df=True*)

获取指定合约或合约列表的历史行情（包含起止日期，日线或分钟线），不能在'handle_bar'函数中进行调用。

注意，这一函数主要是为满足在研究平台编写策略习惯而引入。在编写策略中，使用history_bars进行数据获取会更方便。

Parameters:

- **market_code** (*Union[str, Iterable[str]]*) -- 合约代码，合约代码，可传入market_code, market_code list, symbol, symbol list
- **start_date** (*Union[datetime.date, str]*) -- 开始日期，用户必须指定
- **end_date** (*Optional[Union[datetime.date, datetime.orig_time, str]]*) -- 结束日期，默认为策略当前日期前一天
- **frequency** (*Optional[str]*) -- 历史数据的频率。现在支持日/分钟级别的历史数据，默认为'1d'。使用者可自由选取不同频率，例如'5m'代表5分钟线
- **fields** (*Optional[Iterable[str]]*) -- 期望返回的字段名称，如 open，close 等
- **adjust_type** (*Optional[str]*) -- 股息修复方案。前复权 - pre，后复权 - post，不复权 - none
- **skip_suspended** (*Optional[bool]*) -- 是否跳过停牌数据。默认为False，不跳过，用停牌前数据进行补齐。True则为跳过停牌期。注意，当设置为True时，函数market_code只支持单个合约传入
- **expect_df** (*Optional[bool]*) -- 是否期望始终返回 DataFrame。pandas 0.25.0 以上该参数应设为 True，以避免因试图构建 Panel 产生异常

当 expect_df 为 False 时，返回值的类型如下

- 传入一个market_code，多个fields，函数会返回一个pandas DataFrame
- 传入一个market_code，一个field，函数会返回pandas Series
- 传入多个market_code，一个field，函数会返回一个pandas DataFrame
- 传入多个market_code，函数会返回一个pandas Panel

参数	类型	说明
open	float	开盘价
close	float	收盘价
high	float	最高价
low	float	最低价
high_limited	float	涨停价
low_limited	float	跌停价
total_turnover	float	总成交额
volume	float	总成交量
acc_net_value	float	累计净值（仅限基金日线数据）
unit_net_value	float	单位净值（仅限基金日线数据）
discount_rate	float	折价率（仅限基金日线数据）
settlement	float	结算价（仅限期货日线数据）
prev_settlement	float	昨日结算价（仅限期货日线数据）
open_interest	float	累计持仓量（期货专用）

参数	类型	说明
basis_spread	float	基差点数（股指期货专用，股指期货收盘价-标的指数收盘价）
trading_date	pandas.Timestamp	交易日期（仅期货分钟线数据），对应期货夜盘的情况

Example:

获取单一股票历史日线行情:

```
1 get_price('000001.SZ', start_date='2015-04-01', end_date='2015-04-12')
2 #[Out]
3 #open      close      high      low      total_turnover      volume      high_limited      low_limited
4 #2015-04-01      10.7300      10.8249      10.9470      10.5469      2.608977e+09      236637563.0      11.7542      9.6177
5 #2015-04-02      10.9131      10.7164      10.9470      10.5943      2.222671e+09      202440588.0      11.9102      9.7397
6 #2015-04-03      10.6486      10.7503      10.8114      10.5876      2.262844e+09      206631550.0      11.7881      9.6448
7 #2015-04-07      10.9538      11.4015      11.5032      10.9538      4.898119e+09      426308008.0      11.8288      9.6787
8 #2015-04-08      11.4829      12.1543      12.2628      11.2929      5.784459e+09      485517069.0      12.5409      10.2620
9 #2015-04-09      12.1747      12.2086      12.9208      12.0255      5.794632e+09      456921108.0      13.3684      10.9403
10 #2015-04-10      12.2086      13.4294      13.4294      12.1069      6.339649e+09      480990210.0      13.4294      10.9877
11 #...
```

Return type:

Union[pd.DataFrame, pd.Panel, pd.Series]

get_price_change_rate - 历史涨跌幅

deepquant. quest. api. get_price_change_rate(*market_code, count=1, expect_df=False*)

获取股票/指数截止T-1日的日涨幅

Parameters:

- **market_code** (Union[str, List[str]]) -- 可输入 market_code, market_code list, symbol, symbol list
- **count** (Optional[int]) -- 回溯获取的数据个数。默认为当前能够获取到的最近的数据
- **expect_df** (Optional[bool]) -- 是否期望始终返回 DataFrame。pandas 0.25.0 以上该参数应设为 True，以避免因试图构建 Panel 产生异常

当 expect_df 为 False 时，返回值的类型如下：

- 传入多个market_code，函数会返回pandas DataFrame
- 传入一个market_code，函数会返回pandas Series

Example:

获取平安银行以及沪深300指数一段时间的涨跌幅情况:

```
1 get_price_change_rate(['000001.SZ', '510050.SH'], 1)
2 # [Out]
3 # 2016-06-01 15:30:00.00 INFO market_code 000001.SZ 510050.SH
4 # date
5 # 2016-05-31 0.026265 0.033964
6 # 2016-06-02 15:30:00.00 INFO market_code 000001.SZ 510050.SH
7 # date
8 # 2016-06-01 -0.006635 -0.008308
```

Return type:

Union[DataFrame, Series]

股票

industry - 行业股票列表

deepquant. quest. api. get_industry(*industry, source=12*)

通过传入行业名称、行业指数代码或者行业代号，拿到 T 日指定行业的股票列表

Parameters:

- **industry** (str) -- 行业名字
- **source** (Optional[str]) -- 默认为中证监会(12)，可选万得到(guildata)，中信

Return type:

List[str]

Returns:

list of market_code

sector - 板块股票列表

deepquant. quest. api. sector(*code*)

获得属于某一板块的所有股票列表。

Parameters:

code (str) -- 板块名称或板块代码。例如，能源板块可填写'Energy'、'能源'或sector_code.Energy

目前支持的板块分类如下，其取值参考自MSCI发布的全球行业标准分类:

板块代码	中文板块名称	英文板块名称
Energy	能源	energy
Materials	原材料	materials
ConsumerDiscretionary	非必需消费品	consumer discretionary
ConsumerStaples	必需消费品	consumer staples
HealthCare	医疗保健	health care

板块代码	中文板块名称	英文板块名称
Financials	金融	financials
InformationTechnology	信息技术	information technology
TelecommunicationServices	电信服务	telecommunication services
Utilities	公共服务	utilities
Industrials	工业	industrials

Example:

```
1 def init(context):
2     ids1 = sector("consumer discretionary")
3     ids2 = sector("非必需消费品")
4     ids3 = sector("ConsumerDiscretionary")
5     assert ids1 == ids2 and ids1 == ids3
6     logger.info(ids1)
7 #INIT INFO
8 #['002045.SZ', '603099.SH', '002486.SZ', '002536.SZ', '300100.SZ', '600633.SH', '002291.SZ', ..., '600233.SH']
```

Return type:

List[str]

get_dividend - 获取分红数据[🔗](#)

deepquant.quest.api.get_dividend(*market_code*, *start_date*)[🔗](#)

获取某只股票到策略当前日期前一天的分红情况（包含起止日期）。

Parameters:

- **market_code** (*str*) -- 股票代码
- **start_date** (*Union[str, datetime.date, datetime.orig_time, pd.Timestamp]*) -- 开始日期，需要早于策略当前日期

fields	字段名
announcement_date	分红宣布日
book_closure_date	股权登记日
dividend_cash_before_tax	税前分红
ex_dividend_date	除权除息日
payable_date	分红到帐日
round_lot	分红最小单位

Example:

获取平安银行2013-01-04 到策略当前日期前一天的分红数据:

```
1 get_dividend('000001.SZ', start_date='20130104')
2 #[Out]
3 #array([[20130614, 20130619, 20130620, 20130620, 1.7, 10),
4 #       (20140606, 20140611, 20140612, 20140612, 1.6, 10),
5 #       (20150407, 20150410, 20150413, 20150413, 1.74, 10),
6 #       (20160608, 20160615, 20160616, 20160616, 1.53, 10)],
7 #      dtype=[('announcement_date', '<u4'), ('book_closure_date', '<u4'), ('ex_dividend_date', '<u4'), ('payable_date', '<u4'), ('dividend_cash_before_tax', '<f8'), ('round_lot', '<u4')])
```

Return type:

Optional[np.ndarray]

is_suspended - 全天停牌判断[🔗](#)

deepquant.quest.api.is_suspended(*market_code*)[🔗](#)

判断某只股票是否全天停牌。

Parameters:

- **market_code** (*str*) -- 某只股票的代码或股票代码，可传入单只股票的market_code, symbol
- **count** (*Optional[int]*) -- 回溯获取的数据个数。默认为当前能够获取到的最近的数据

Return type:

Union[bool, DataFrame]

is_st_stock - ST股判断[🔗](#)

deepquant.quest.api.is_st_stock(*market_code*)[🔗](#)

判断股票在一段时间内是否为ST股（包括ST与*ST）。

ST股是有退市风险因此风险比较大的股票，很多时候您也会希望判断自己使用的股票是否是'ST'股来避开这些风险大的股票。另外，我们目前的策略比赛也禁止了使用'ST'股。

Parameters:

- **market_code** (*str*) -- 某只股票的代码，可传入单只股票的market_code, symbol
- **count** (*Optional[int]*) -- 回溯获取的数据个数。默认为当前能够获取到的最近的数据

Return type:

Union[bool, DataFrame]

get_split - 拆分数据[🔗](#)

deepquant. quest. api. get_split(*market_code*, *start_date=None*)[¶](#)

获取某只股票到策略当前日期前一天的拆分情况（包含起止日期）。

Parameters:

- **market_code** (Union[str, List[str]]) -- 证券代码，证券的独特的标识符，例如：'000001.SZ'
- **start_date** (Union[str, date, None]) -- 开始日期，用户必须指定，需要早于策略当前日期

Return type:

DataFrame

Returns:

查询时间段内的某个股票的拆分数据

- ex_dividend_date: 除权除息日，该天股票的价格会因为拆分而进行调整
- book_closure_date: 股权登记日
- split_coefficient_from: 拆分子（拆分前）
- split_coefficient_to: 拆分子（拆分后）

例如：每10股转增2股，则split_coefficient_from = 10, split_coefficient_to = 12.

Example:

```
1 get_split('000001.SZ', start_date='2010-01-04')
2 #[Out]
3 #
4 #2013-06-20      book_closure_date payable_date  split_coefficient_from      #ex_dividend_date
5 #
6 #ex_dividend_date      split_coefficient_to
7 #2013-06-20
                                16.0
```

get_securities_margin - 融资融券信息[¶](#)

deepquant. quest. api. get_securities_margin(*market_code*, *count=1*, *fields=None*, *expect_df=False*)[¶](#)

获取融资融券信息。包括 [深证融资融券数据](#) 以及 [上证融资融券数据](#) 情况。既包括个股数据，也包括市场整体数据。需要注意，融资融券的开始日期为2010年3月31日。

Parameters:

- **market_code** (Union[str, Iterable[str]]) -- 可输入market_code, market_code list, symbol, symbol list。另外，输入'XSHG'或'sh'代表整个上证整体情况；'XSHE'或'sz'代表深证整体情况
- **count** (Optional[int]) -- 回溯获取的数据个数。默认为当前能够获取到的最近的数据
- **fields** (Optional[str]) -- 期望返回的字段，默认为所有字段。见下方列表
- **expect_df** (Optional[bool]) -- 是否期望始终返回 DataFrame。pandas 0.25.0 以上该参数应设为 True，以避免因试图构建 Panel 产生异常

fields	字段名
margin_balance	融资余额
buy_on_margin_value	融资买入额
margin_repayment	融资偿还额
short_balance	融券余额
short_balance_quantity	融券余量
short_sell_value	融券卖出额
short_sell_quantity	融券卖出量
short_repayment_quantity	融券偿还量
total_balance	融资融券余额

当 expect_df 为 False 时，返回值的类型如下：

- 多个market_code，单个field的时候返回DataFrame，index为date，column为market_code
- 单个market_code，多个fields的时候返回DataFrame，index为date，column为fields
- 单个market_code，单个field返回Series
- 多个market_code，多个fields的时候返回DataPanel Items axis为fields Major_axis axis为时间戳 Minor_axis axis为market_code

Example:

- 获取沪深两个市场一段时间内的融资余额:

```
1 logger.info(get_securities_margin('510050.SH', count=5))
2 #[Out]
3 #margin_balance  buy_on_margin_value  short_sell_quantity  margin_repayment  short_balance_quantity  short_repayment_quantity  short_balance  total_balance
4 #2016-08-01      7.811396e+09      50012306.0      3597600.0      41652042.0      15020600.0      1645576.0      NaN      NaN
5 #2016-08-02      7.826381e+09      34518238.0      2375700.0      19532586.0      14154000.0      3242300.0      NaN      NaN
6 #2016-08-03      7.733306e+09      17967333.0      4719700.0      111043009.0      16235600.0      2638100.0      NaN      NaN
7 #2016-08-04      7.741497e+09      30259359.0      6488600.0      22068637.0      17499000.0      5225200.0      NaN      NaN
8 #2016-08-05      7.726343e+09      25270756.0      2865863.0      40423859.0      14252363.0      6112500.0      NaN      NaN
```

- 获取沪深两个市场一段时间内的融资余额:

```
1 logger.info(get_securities_margin(['SZ', 'SH'], count=5, fields='margin_balance'))
2 #[Out]
3 #      SZ      SH
4 #2016-08-01      3.837627e+11      4.763557e+11
5 #2016-08-02      3.828923e+11      4.763931e+11
6 #2016-08-03      3.823545e+11      4.769321e+11
7 #2016-08-04      3.833260e+11      4.776380e+11
8 #2016-08-05      3.812751e+11      4.766928e+11
```

- 获取上证个股以及整个上证市场融资融券情况:


```
1 logger.info(get_securities_margin(['SH', '601988.SH', '510050.SH'], count=5))
2 #[Out]
3 #<class 'pandas.core.panel.Panel'>
4 #Dimensions: 8 (items) x 5 (major_axis) x 3 (minor_axis)
5 #Items axis: margin_balance to total_balance
6 #Major_axis axis: 2016-08-01 00:00:00 to 2016-08-05 00:00:00
7 #Minor_axis axis: SH to 510050.SH
```

- 获取50ETF融资偿还额情况

```
1 logger.info(get_securities_margin('510050.SH', count=5, fields='margin_repayment'))
2 #[Out]
3 #2016-08-01      41652042.0
4 #2016-08-02      19532586.0
5 #2016-08-03      111043009.0
6 #2016-08-04      22068637.0
7 #2016-08-05      40423859.0
8 #Name: margin_repayment, dtype: float64
```

Return type:

Union[pd.Series, pd.DataFrame, pd.Panel]

concept - 概念股列表

deepquant.quest.api.concept(**concept_names*)

获取T日的概念股列表

Parameters:

- concept_names** (str) -- 概念名称。可以从概念列表中选择一个或多个概念填写, 可以通过 deepquant.quest.datac.concept_list() 获取概念列表

Return type:

List[str]

Returns:

属于该概念的股票 market_code

Example:

得到一个概念的股票列表:

```
1 concept(' 民营医院')
2 # [Out]
3 # ['600105.SH',
4 #  '002550.SZ',
5 #  '002004.SZ',
6 #  '002424.SZ',
7 #  ...]
```

得到某几个概念的股票列表:

```
1 concept(' 民营医院', ' 国企改革')
2 # [Out]
3 # ['601607.SH',
4 #  '600748.SH',
5 #  '600630.SH',
6 #  ...]
```

get_margin_stocks - 融资融券列表

deepquant.quest.api.get_margin_stocks(*exchange=None, margin_type='all'*)

获取某个日期深证、上证融资融券股票列表。

Parameters:

- exchange** (Optional[str]) -- 交易所，默认为 None，返回所有字段。可选字段包括: 'XSHE', 'sz' 代表深交所; 'XSHG', 'sh' 代表上交所
- margin_type** (str) -- 'stock' 代表融券卖出, 'cash', 代表融资买入, 'all', 代表包含融资和融券，默认为'all'

Return type:

List[str]

Returns:

属于该概念的股票 market_code

Example:

获取沪深市场的融券卖出列表:

```
1 get_margin_stocks(exchange=None, margin_type='stock')
2 # [Out]
3 # ['000001.SZ',
4 #  '000002.SZ',
5 #  '000006.SZ',
6 #  ...]
```

获取沪深市场融资买入列表:

```
1 get_margin_stocks(exchange='SZ', margin_type='stock')
2 # [Out]
3 # ['000001.SZ',
4 #  '000002.SZ',
5 #  '000006.SZ',
6 #  ...]
```

获取上证融资买入列表:

```
1 get_margin_stocks(exchange='SH', margin_type='cash')
2 # [Out]
3 # ['510050.SH',
4 #  '510160.SH',
5 #  '510180.SH',
6 #  ...]
```

get_shares - 流通股信息

deepquant.quest.api.get_shares(*market_code, count=1, fields=None, expect_df=False*)

Parameters:

- market_code** (Union[str, List[str]]) -- 可输入 market_code, market_code list, symbol, symbol list

- **count** (Optional[int]) -- 回溯获取的数据个数。默认为当前能够获取到的最近的数据
- **fields** (Optional[str]) -- 期望返回的字段，默认为所有字段。见下方列表
- **expect_df** (Optional[bool]) -- 是否期望始终返回 DataFrame。pandas 0.25.0 以上该参数应设为 True，避免因试图构建 Panel 产生异常

fields	字段名
total	总股本
circulation_a	流通A股
management_circulation	已流通高管持股
non_circulation_a	非流通A股合计
total_a	A股总股本

Return type:

Union[DataFrame, Series]

Returns:

查询时间段内某个股票的流通情况，当 expect_df 为 False 且 fields 指定为单一字段的情况时返回 pandas.Series

Example:

获取平安银行总股本数据:

```
1 logger.info(get_shares('000001.SZ', count=5, fields='total'))
2 #[Out]
3 #2016-08-01    1.717041e+10
4 #2016-08-02    1.717041e+10
5 #2016-08-03    1.717041e+10
6 #2016-08-04    1.717041e+10
7 #2016-08-05    1.717041e+10
8 #Name: total, dtype: float64
```

get_turnover_rate - 历史换手率¹

deepquant.quest.api.get_turnover_rate(*market_code*, *count*=1, *fields*=None, *expect_df*=False)¹

获取截止T-1交易日的换手率数据

param market_code:

可输入 market_code, market_code list, symbol, symbol list

param count:

回溯获取的数据个数。默认为当前能够获取到的最近的数据

param fields:

期望返回的字段，默认为所有字段。见下方列表

param expect_df:

是否期望始终返回 DataFrame。pandas 0.25.0 以上该参数应设为 True，避免因试图构建 Panel 产生异常

fields	字段名
today	当天换手率
week	过去一周平均换手率
month	过去一个月平均换手率
three_month	过去三个月平均换手率
six_month	过去六个月平均换手率
year	过去一年平均换手率
current_year	当年平均换手率
total	上市以来平均换手率

当 expect_df 为 False 时，返回值的类型如下：

- 如果只传入一个market_code，多个fields，返回 *pandas.DataFrame*
- 如果传入market_code list，并指定单个field，函数会返回一个 *pandas.DataFrame*
- 如果传入market_code list，并指定多个fields，函数会返回一个 *pandas.Panel*

example:

获取平安银行历史换手率情况:

```
1 logger.info(get_turnover_rate('000001.SZ', count=5))
2 #[Out]
3 #          today    week    month    three_month    six_month    year          #2016-08-01    0.5190    0.4478    0.3213          0.2877          0.3442    0.5027
4 #2016-08-02    0.3070    0.4134    0.3112          0.2843    0.3427    0.5019
5 #2016-08-03    0.2902    0.3460    0.3102          0.2823    0.3432    0.4982
6 #2016-08-04    0.9189    0.4938    0.3331          0.2914    0.3482    0.4992
7 #2016-08-05    0.4962    0.5031    0.3426          0.2960    0.3504    0.4994
8
9 #          current_year    total
10 #2016-08-01          0.3585    1.1341
11 #2016-08-02          0.3570    1.1341
12 #2016-08-03          0.3565    1.1339
13 #2016-08-04          0.3604    1.1339
14 #2016-08-05          0.3613    1.1338
```

Return type:

Union[pd.Series, pd.DataFrame, pd.Panel]

get_factor - 因子

deepquant. quest. api. get_factor(*market_code, factors, count=1, universe=None, expect_df=False*)

获取股票截止T-1日的因子数据

Parameters:

- **market_code** (*Union[str, List[str]]*) -- 合约代码，可传入market_code, market_code list
- **factors** (*Union[str, List[str]]*) -- 因子名称，可查询 deepquant.quest.datac.get_all_factor_names() 得到所有有效因子字段
- **count** (*Optional[int]*) -- 获取多少个交易日的数据
- **universe** (*Optional[Union[str, List[Union]]]*) -- 当获取横截面因子时，universe指定了因子计算时的股票池
- **expect_df** (*Optional[bool]*) -- 默认为False。当设置为True时，总是返回 multi-index DataFrame。pandas 0.25.0 以上该参数应设为 True，以避免因试图构建 Panel 产生异常

Return type:

pd.DataFrame

get_industry - 行业股票列表

deepquant. quest. api. get_industry(*industry, source=12*)

通过传入行业名称、行业指数代码或者行业代号，拿到 T 日指定行业的股票列表

Parameters:

- **industry** (*str*) -- 行业名字
- **source** (*Optional[str]*) -- 默认为中证监会(12)，可选万得到(guildata)，中信

Return type:

List[str]

Returns:

list of market_code

get_instrument_industry - 股票行业分类

deepquant. quest. api. get_instrument_industry(*market_code, source='14', level=1*)

获取T日时股票行业分类

Parameters:

market_code (Union[str, List[str]]) -- 合约代码，可传入market_code, market_code list

:param source:分类来源。'10':万得行业分类 '12': 证监会行业分类 '13':申万行业分类 '14':中信行业分类 默认'14':type level: Optional[int]:param level: 默认为1，可选 0 1 2 3，0表示返回所有级别

Return type:

DataFrame

get_stock_connect - 沪深港通持股信息

deepquant. quest. api. get_stock_connect(*market_code, count=1, fields=None, expect_df=False*)

获取截止T-1日A股股票在香港上市交易的持股情况

Parameters:

- **market_code** (Union[str, List[str]]) -- 合约代码，可传入market_code, market_code list，这里输入的是A股编码
- **count** (Optional[int]) -- 向前获取几个交易日
- **fields** (Optional[str]) -- 持股量 (shares_holding)，持股比例 (holding_ratio)，默认为所有字段
- **expect_df** (Optional[bool]) -- 默认为False。当设置为True时，总是返回 multi-index DataFrame。pandas 0.25.0 以上该参数应设为 True，以避免因试图构建 Panel 产生异常

当 expect_df 为 False 时，返回值的类型如下：

- 多个market_code，多个fields的时候返回pandas Panel
- 单个market_code，多个fields的时候返回pandas DataFrame
- 单个market_code，单个field返回pandas Series

Return type:

DataFrame

Returns:

current_performance - 财务快报数据

deepquant. quest. api. current_performance(*market_code, info_date=None, quarter=None, interval='1q', fields=None*)

默认返回给定的 market_code 当前最近一期的快报数据

Parameters:

- **market_code** (*str*) -- 合约代码
- **info_date** (*Optional[str]*) -- yyyymmdd 或者 yyyy-mm-dd。如果不填(info_date和quarter都为空)，则返回策略运行当前日期的最新发布的快报。如果填写，则从info_date当天或者之前最新的报告开始抓取。

- **quarter** (*Optional[str]*) -- info_date参数优先级高于quarter。如果info_date填写了日期，则不查看quarter这个字段。 如果info_date没有填写而quarter 有填写，则财报回溯查询的起始报告期，例如'2015q2', '2015q4'分别代表2015年半年报以及年报。默认只获取当前报告期财务信息
- **interval** (*Optional[str]*) -- 查询财务数据的间隔。例如， 填写'5y'， 则代表从报告期开始回溯5年， 每年为相同报告期数据； '3q'则代表从报告期开始向前回溯3个季度。不填写默认抓取一期。

- **fields** (*Optional[str, List[str]]*) -- 抓取对应有效字段返回。默认返回所有字段。

Return type:

pd.DataFrame

get_pit_financials_ex - 季度财务信息

deepquant. quest. api. get_pit_financials_ex(*market_code, fields, count, statements='latest'*)

以给定一个报告期回溯的方式获取季度基础财务数据（三大表），即利润表（income_statement），资产负债表（balance_sheet），现金流量表（cash_flow_statement）。

Parameters:

- **market_code** (*Union[str, List[str]]*) -- 合约代码，可传入market_code, market_code list，这里输入的是A股编码
- **fields** (*Union[str, List[str]]*) -- 需要返回的财务字段
- **count** (*int*) -- 几条数据
- **statements** (*str*) -- 设置 statements 为 all 时返回所有记录，statements 等于 latest 时返回最新的一条记录，默认为 latest.

Return type:

Optional[DataFrame]

Returns:

指数

index_components - 指数成分股

deepquant. quest. api. index_components(*market_code, date=None*)

获取某一指数的股票构成列表，也支持指数的历史构成查询。

Parameters:

- **market_code** (*str*) -- 指数代码，可传入market_code
- **date** (*Union[str, date, None]*) -- 查询日期，默认为策略当前日期。如指定，则应保证该日期不晚于策略当前日期

Return type:

List[str]

Returns:

构成该指数股票的 market_code

Example:

得到上证指数在策略当前日期的构成股票的列表:

```
1 index_components('000001.SH')
2 #[Out][ '600000.SH', ' 600004.SH', ...]
```

index_weights - 指数成分股权重

deepquant. quest. api. index_weights(*market_code, date=None*)

获取T-1日的指数权重

Parameters:

- **market_code** (*str*) -- 指数
- **date** (*Union[str, date, None]*) -- 可选，默认为T-1日

Return type:

Series

Returns:

每只股票在指数中的构成权重

Example:

获取上证50指数上个交易日的指数构成

```
1 index_weights('000016.SH')
2 # [Out]
3 # market_code
4 # 600000.SH      0.03750
5 # 600010.SH      0.00761
6 # 600016.SH      0.05981
7 # 600028.SH      0.01391
8 # 600029.SH      0.00822
9 # 600030.SH      0.03526
10 # 600036.SH      0.04889
11 # 600050.SH      0.00998
12 # 600104.SH      0.02122
```

宏观经济

econ.get_reserve_ratio - 存款准备金率

deepquant. quest. api. econ. get_reserve_ratio(*reserve_type='all', n=1*)

获取截止T日的存款准备金率

Parameters:

- **reserve_type** (*str*) -- 目前有大型金融机构（'major'）和其他金融机构（'other'）两种分类。默认为all，即所有类别都会返回。
- **n** (*int*) -- 返回最近 n 个生效日期的存款准备金率数据

Return type:

Optional[DataFrame]

[Next](#) [Previous](#)

