

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
 - OrderStyle - 订单类型
 - submit_order - 自由参数下单「通用」
 - order - 智能下单「通用」
 - order_to - 智能下单「通用」
 - order_shares - 指定股数交易
 - order_lots - 指定手数交易
 - order_value - 指定价值交易
 - order_percent - 一定比例下单
 - order_target_value - 目标价值下单
 - order_target_percent - 目标比例下单
 - order_target_portfolio - 批量调仓
 - buy_open(期货期权) - 买开
 - sell_close(期货期权) - 平买仓
 - sell_open(期货期权) - 卖开
 - buy_close(期货期权) - 平卖仓
 - exercise(期权/转债) - 行权
 - cancel_order - 撤单
 - get_open_orders - 获取未成交订单数据
 - subscribe_value - 基金申购指令
 - subscribe_percent - 按权重申购基金
 - subscribe_shares - 按照份额申购
 - redeem_percent - 按权重赎回基金
 - redeem_shares - 按份额赎回基金
 - redeem_value - 根据赎回总金额计算份额
- 仓位查询接口
- 数据查询接口
- 其他接口
- scheduler 定时器
- 类
- 枚举常量

银河启睿策略中心

- »
- 交易接口
- [View page source](#)

交易接口

OrderStyle - 订单类型

该类型可供后续下单接口中 price_or_style 参数使用

```
class deepquant.quest.model.order.MarketOrder

    order_shares("000001.SZ", amount=100, price_or_style=MarketOrder())
```

市价单

```
class deepquant.quest.model.order.LimitOrder(limit_price)
```

Parameters:

limit_price (float) -- 价格

```
order_shares("000001.SZ", amount=100, price_or_style=LimitOrder(10))
```

限价单

```
class deepquant.quest.model.order.TWAPOrder(start_min, end_min)
```

Parameters:

- start_min (int) -- 分钟起始时间
- end_min (int) -- 分钟结束时间

```
order_shares("000001.SZ", amount=100, price_or_style=TWAPOrder(931, 945))
```

算法时间加权价格订单

```
class deepquant.quest.model.order.VWAPOrder(start_min, end_min)
```

Parameters:

- **start_min** (*int*) -- 分钟起始时间
- **end_min** (*int*) -- 分钟结束时间

```
order_shares("000001.SZ", amount=100, price_or_style=VWAPOrder(931, 945))
```

算法成交量加权价格订单

submit_order - 自由参数下单「通用」[🔗](#)

deepquant. quest. api. submit_order(*id_or_ins, amount, side, price=None, position_effect=None*)[🔗](#)

通用下单函数，策略可以通过该函数自由选择参数下单。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (float) -- 下单量，需为正数
- **side** (SIDE) -- 多空方向
- **price** (Optional[float]) -- 下单价格，默认为None，表示市价单
- **position_effect** (Optional[POSITION_EFFECT]) -- 开平方向，交易股票不需要该参数

Example:

```
# 购买 2000 股的平安银行股票，并以市价单发送：
submit_order('000001.SZ', 2000, SIDE.BUY)
# 平 10 份 RB1812 多方向的今仓，并以 4000 的价格发送限价单
submit_order('RB1812', 10, SIDE.SELL, price=4000, position_effect=POSITION_EFFECT.CLOSE_TODAY)
```

Return type:

Optional[[Order](#)]

order - 智能下单「通用」[🔗](#)

deepquant. quest. api. order(*market_code, quantity, price_or_style=None, price=None, style=None*)[🔗](#)

全品种通用智能调仓函数

如果不指定 price, 则相当于下 MarketOrder

如果 market_code 是股票，等同于调用 order_shares

如果 market_code 是期货，则进行智能下单:

- quantity 表示调仓量
- 如果 quantity 为正数，则先平 Sell 方向仓位，再开 Buy 方向仓位
- 如果 quantity 为负数，则先平 Buy 反向仓位，再开 Sell 方向仓位

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 下单标的物
- **quantity** (int) -- 调仓量
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
1 # 当前仓位为0
2 # RB1710 多方向调仓2手：调整后变为 BUY 2手
3 order('RB1710', 2)
4
5 # RB1710 空方向调仓3手：先平多方向2手 在开空方向1手，调整后变为 SELL 1手
6 order('RB1710', -3)
```

Return type:

List[[Order](#)]

order_to - 智能下单「通用」[🔗](#)

deepquant. quest. api. order_to(*market_code, quantity, price_or_style=None, price=None, style=None*)[🔗](#)

全品种通用智能调仓函数

如果不指定 price, 则相当于 MarketOrder

如果 market_code 是股票，则表示仓位调整到多少股

如果 market_code 是期货，则进行智能调仓:

- quantity 表示调整至某个仓位
- quantity 如果为正数，则先平 SELL 方向仓位，再 BUY 方向开仓 quantity 手
- quantity 如果为负数，则先平 BUY 方向仓位，再 SELL 方向开仓 -quantity 手

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 下单标的物
- **quantity** (*int*) -- 调仓量
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
1 # 当前仓位为0
2 # RB1710 调仓至 BUY 2手
3 order_to('RB1710', 2)
4
5 # RB1710 调仓至 SELL 1手
6 order_to('RB1710', -1)
```

Return type:

List[[Order](#)]

order_shares - 指定股数交易[¶](#)

deepquant. quest. api. order_shares(*id_or_ins, amount, price_or_style=None, price=None, style=None*)[¶](#)

指定股数的买/卖单，最常见的落单方式之一。如有需要落单类型当做一个参量传入，如果忽略掉落单类型，那么默认是市价单（market order）。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (*int*) -- 下单量, 正数代表买入，负数代表卖出。将会根据一手xx股来向下调整到一手的倍数，比如中国A股就是调整成100股的倍数。
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#购买Buy 2000 股的平安银行股票，并以市价单发送：
order_shares('000001.SZ', 2000)
#卖出2000股的平安银行股票，并以市价单发送：
order_shares('000001.SZ', -2000)
#购买1000股的平安银行股票，并以限价单发送，价格为¥11：
order_shares('000001.SZ', 1000, price_or_style=11)
#购买1000股的平安银行股票，并以限价单发送，价格为¥10：
order_shares('000001.SZ', 1000, price_or_style=LimitOrder(10))
#购买1000股的平安银行股票，并以 9:31 到 9:45 的VWAP价格发送：
order_shares('000001.SZ', 1000, price_or_style=VWAPOrder(931, 945))
```

Return type:

Optional[[Order](#)]

order_lots - 指定手数交易[¶](#)

deepquant. quest. api. order_lots(*id_or_ins, amount, price_or_style=None, price=None, style=None*)[¶](#)

指定手数发送买/卖单。如有需要落单类型当做一个参量传入，如果忽略掉落单类型，那么默认是市价单（market order）。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (*int*) -- 下单量, 正数代表买入，负数代表卖出。将会根据一手xx股来向下调整到一手的倍数，比如中国A股就是调整成100股的倍数。
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#买入20手的平安银行股票，并且发送市价单：
order_lots('000001.SZ', 20)
#买入10手平安银行股票，并且发送限价单，价格为¥10：
order_lots('000001.SZ', 10, price_or_style=LimitOrder(10))
```

Return type:

Optional[[Order](#)]

order_value - 指定价值交易[¶](#)

deepquant. quest. api. order_value(*id_or_ins, cash_amount, price_or_style=None, price=None, style=None*)[¶](#)

使用想要花费的金钱买入/卖出股票，而不是买入/卖出想要的股数，正数代表买入，负数代表卖出。股票的股数总是会被调整成对应的100的倍数（在A中国A股市场1手是100股）。 如果资金不足，该API将会使用最大可用资金发单。

需要注意： 当您提交一个买单时，cash_amount 代表的含义是您希望买入股票消耗的金额（包含税费），最终买入的股数不仅和发单的价格有关，还和税费相关的参数设置有关。 当您提交一个卖单时，cash_amount 代表的意义是您希望卖出股票的总价值。如果金额超出了您所持有股票的价值，那么您将卖出所有股票。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **cash_amount** (float) -- 需要花费现金购买/卖出证券的数目。正数代表买入，负数代表卖出。
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#花费最多¥10000买入平安银行股票，并以市价单发送。具体下单的数量与您策略税费相关的配置有关。
order_value('000001.SZ', 10000)
#卖出价值¥10000的现在持有的平安银行，以10¥价格发出限价单：
order_value('000001.SZ', -10000, price_or_style=10)
```

Return type:

Optional[[Order](#)]

order_percent - 一定比例下单[¶](#)

deepquant. quest. api. order_percent(*id_or_ins, percent, price_or_style=None, price=None, style=None*)[¶](#)

发送一个花费价值等于目前投资组合（市场价值和目前现金的总和）一定百分比现金的买/卖单，正数代表买，负数代表卖。股票的股数总是会被调整成对应的一手的股票数的倍数（1手是100股）。百分比是一个小数，并且小于或等于1（<=100%），0.5表示的是50%.需要注意，如果资金不足，该API将会使用最大可用资金发单。

需要注意：

发送买单时，percent 代表的是期望买入股票消耗的金额（包含税费）占投资组合总权益的比例。 发送卖单时，percent 代表的是期望卖出的股票总价值占投资组合总权益的比例。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **percent** (float) -- 占有现有的投资组合价值的百分比。正数表示买入，负数表示卖出。
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#花费等于现有投资组合50%价值的现金买入平安银行股票：
order_percent('000001.SH', 0.5)
#花费等于现有投资组合50%价值的现金买入平安银行股票，以10¥限价单：
order_percent('000001.SH', 0.5, price_or_style=10)
```

Return type:

Optional[[Order](#)]

order_target_value - 目标价值下单[¶](#)

deepquant. quest. api. order_target_value(*id_or_ins, cash_amount, price_or_style=None, price=None, style=None*)[¶](#)

买入/卖出并且自动调整该证券的仓位到一个目标价值。 加仓时，cash_amount 代表现有持仓的价值加上即将花费（包含税费）的现金的总价值。 减仓时，cash_amount 代表调整仓位的目标价至。

需要注意，如果资金不足，该API将会使用最大可用资金发单。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **cash_amount** (float) -- 最终的该证券的仓位目标价值。
- **price_or_style** (Union[float, OrderStyle, None, Tuple, Tuple[Union[int, float, OrderStyle, None]], Tuple[Union[int, float, OrderStyle, None], Union[int, float, OrderStyle, None]]]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#如果现在的投资组合中持有价值¥3000的平安银行股票的仓位，以下代码范例会发送花费 ¥7000 现金的平安银行买单到市场。（向下调整到最接近每手股数即100的倍数的股数）：
order_target_value('000001.SZ', 10000)
#如果现在的投资组合中持有价值¥3000的平安银行股票的仓位，以下代码范例会发送10¥限价单共花费 ¥7000 现金的平安银行买单到市场
#或者如果现在的投资组合中持有价值¥13000的平安银行股票的仓位，以下代码范例会发送11¥限价单共花费 ¥3000 现金的平安银行卖单到市场
order_target_value('000001.SZ', 10000, price_or_style=(10, 11))
```

Return type:

Optional[\[Order\]](#)

order_target_percent - 目标比例下单[¶](#)

deepquant. quest. api. order_target_percent(*id_or_ins, percent, price_or_style=None, price=None, style=None*)[¶](#)

买入/卖出证券以自动调整该证券的仓位到占有一个目标价值。

加仓时，percent 代表证券已有持仓的价值加上即将花费的现金（包含税费）的总值占当前投资组合总价值的比例。 减仓时，percent 代表证券将被调整到的目标价至占当前投资组合总价值的比例。

其实我们需要计算一个position_to_adjust (即应该调整的仓位)

position_to_adjust = target_position - current_position

投资组合价值等于所有已有仓位的价值和剩余现金的总和。买/卖单会被下舍入一手股数（A股是100的倍数）的倍数。目标百分比应该是一个小数，并且最大值应该<=1，比如0.5表示50%。

如果 position_to_adjust 计算之后是正的，那么会买入该证券，否则会卖出该证券。需要注意，如果需要买入证券而资金不足，该 API 将使用最大可用资金发出订单。

另外，如果您希望大量调整股票仓位，推荐使用 order_target_portfolio 而非在循环中调取 order_target_percent，前者将拥有更好的性能。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **percent** (float) -- 仓位最终所占投资组合总价值的目标百分比。
- **price_or_style** (Union[float, OrderStyle, None, Tuple, Tuple[Union[int, float, OrderStyle, None]], Tuple[Union[int, float, OrderStyle, None], Union[int, float, OrderStyle, None]]]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#如果投资组合中已经有了平安银行股票的仓位，并且占据目前投资组合的10%的价值，那么以下代码会消耗相当于当前投资组合价值5%的现金买入平安银行股票：
order_target_percent('000001.SZ', 0.15)
#如果投资组合中已经有了平安银行股票的仓位，并且占据目前投资组合的10%的价值，那么以下代码会消耗相当于当前投资组合价值5%的现金以10¥限价单买入平安银行股票：
#或者如果投资组合中已经有了平安银行股票的仓位，并且占据目前投资组合的20%的价值，那么以下代码会消耗相当于当前投资组合价值5%的现金以11¥限价单卖出平安银行股票：
order_target_percent('000001.SZ', 0.15, price_or_style=(10, 11))
```

Return type:

Optional[\[Order\]](#)

order_target_portfolio - 批量调仓[¶](#)

deepquant. quest. api. order_target_portfolio(*target_portfolio, price_or_styles={}*)[¶](#)

批量调整股票仓位至目标权重。注意：股票账户中未出现在 target_portfolio 中的资产将被平仓！

该 API 的参数 target_portfolio 为字典，key 为 market_code 或 instrument，value 为权重。 此时将根据参数 price_or_styles 中设置的价格来计算目标持仓数量并调仓。

Parameters:

- **target_portfolio** (Dict[str, float]) -- 目标权重字典，key 为 market_code，value 为权重。
- **price_or_styles** (Dict[str, Union[float, OrderStyle, None, Tuple, Tuple[Union[int, float, OrderStyle, None]], Tuple[Union[int, float, OrderStyle, None], Union[int, float, OrderStyle, None]]]) -- 目标下单价格字典，key 为 market_code，value 为价格或订单类型或订单类型和价格组成的 tuple

Example:

```
# 调整仓位，以使平安银行和万科 A 的持仓占比分别达到 10% 和 15%，同时发送市价单
order_target_portfolio({
    '000001.SZ': 0.1,
    '000002.SZ': 0.15
})

# 调整仓位，分别以 14 和 26 元发出限价单，目标是使平安银行和万科 A 的持仓占比分别达到 10% 和 15%
order_target_portfolio({
    '000001.SZ': 0.1,
    '000002.SZ': 0.15
}, {
    '000001.SZ': 14,
    '000002.SZ': 26,
})

# 调整仓位，使平安银行和万科 A 的持仓占比分别达到 10% 和 15%。
# 其中平安银行的平仓价为 14 元，开仓价为 15 元；万科 A 的平仓价为 26 元，开仓价为 27 元。
order_target_portfolio({
    '000001.SZ': 0.1,
    '000002.SZ': 0.15
}, {
    '000001.SZ': (15, 14),
    '000002.SZ': (27, 26)
})
```

Return type:

List[\[Order\]](#)

buy_open(期货期权) - 买开[¶](#)

deepquant. quest. api. buy_open(*id_or_ins, amount, price_or_style=None, price=None, style=None*)[¶](#)

买入开仓。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (int) -- 下单手数
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
#以价格为3500的限价单开仓买入2张上期所AG1607合约：
buy_open('AG1607', amount=2, price_or_style=3500))
```

Return type:

Union[[Order](#), List[[Order](#)], None]

sell_close(期货期权) - 平买仓[¶](#)

deepquant. quest. api. sell_close(*id_or_ins, amount, price_or_style=None, price=None, style=None, close_today=False*)[¶](#)

平买仓

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (float) -- 下单手数
- **close_today** (Optional[bool]) -- 是否指定发平今仓单，默认为False，发送平仓单
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
# 以市价单单将现有IF1603买入平仓2张：
sell_close('IF1603', 2, price_or_style=MarketOrder())
```

Return type:

Union[[Order](#), List[[Order](#)], None]

sell_open(期货期权) - 卖开[¶](#)

deepquant. quest. api. sell_open(*id_or_ins, amount, price_or_style=None, price=None, style=None*)[¶](#)

卖出开仓

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (int) -- 下单手数
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder

Example:

```
# 以3100发出限价单将现有IF1603卖出开仓2张：
sell_open('IF1603', 2, price_or_style=3100)
```

Return type:

Union[[Order](#), List[[Order](#)], None]

buy_close(期货期权) - 平卖仓[¶](#)

deepquant. quest. api. buy_close(*id_or_ins, amount, price_or_style=None, price=None, style=None, close_today=False*)[¶](#)

平卖仓

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 下单标的物
- **amount** (int) -- 下单手数
- **price_or_style** (Union[int, float, OrderStyle, None]) -- 默认为None，表示市价单，可设置价格，表示限价单，也可以直接设置订单类型，有如下选项：MarketOrder、LimitOrder、TWAPOrder、VWAPOrder
- **close_today** (Optional[bool]) -- 是否指定发平今仓单，默认为False，发送平仓单

Example:

```
#市价单将现有IF1603空仓买入平仓2张:
buy_close(' IF1603', 2)
```

Return type:

Union[[Order](#), List[[Order](#)], None]

exercise(期权/转债) - 行权¹

deepquant. quest. api. exercise(*id_or_ins, amount, convert=False*)¹

行权。针对期权、可转债等含权合约，行使合约权利方被赋予的权利。

Parameters:

- **id_or_ins** (Union[str, [Instrument](#)]) -- 行权合约，market_code 或 Instrument 对象
- **amount** (int) -- 参与行权的合约数量
- **convert** (Optional[bool]) -- 是否为转股（转债行权时可用）

Example:

```
# 行使一张豆粕1905购2350的权力
exercise("M1905C2350", 1)
```

Return type:

Optional[[Order](#)]

cancel_order - 撤单¹

deepquant. quest. api. cancel_order(*order*)¹

撤单

Parameters:

- **order** ([Order](#)) -- 需要撤销的order对象

Return type:

[Order](#)

get_open_orders - 获取未成交订单数据¹

deepquant. quest. api. get_open_orders()¹

获取当日未成交订单数据

Return type:

List[[Order](#)]

subscribe_value - 基金申购指令¹

deepquant. quest. mod. mod_fund. api. subscribe_value(*market_code, cash_amount*)¹

按申购金额申购基金

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 申购基金
- **cash_amount** (float) -- 申购所用资金

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 申购五千块的'华夏成长混合'(000001)
order = subscribe_value('000001', 5000)
assert order.avg_price * order.quantity + order.transaction_cost <= 5000
```

subscribe_percent - 按权重申购基金¹

deepquant. quest. mod. mod_fund. api. subscribe_percent(*market_code, percent*)¹

按账户资金百分比申购基金

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 申购基金
- **percent** (float) -- 可用资金权重

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 用当前账户资金的20%申购
order = subscribe_percent('000001', 0.2)
```

subscribe_shares - 按照份额申购[¶](#)

deepquant.quest.mod.mod_fund.api.subscribe_shares(*market_code, shares*)[¶](#)

按照份额申购

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 申购基金
- **shares** (int) -- 份额

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 申购3000份'华夏成长混合'(000001)
order = subscribe_shares('000001', 3000)
```

redeem_percent - 按权重赎回基金[¶](#)

deepquant.quest.mod.mod_fund.api.redeem_percent(*market_code, percent*)[¶](#)

按权重赎回基金，需要根据赎回总金额，计算份额。按比例赎回时，不足0.01份则不赎回

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 赎回基金
- **percent** (float) -- 可赎回份额权重

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 赎回20%的'华夏成长混合'(000001)
order = redeem_percent('000001', 0.2)
```

redeem_shares - 按份额赎回基金[¶](#)

deepquant.quest.mod.mod_fund.api.redeem_shares(*market_code, shares*)[¶](#)

按份额赎回基金

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 赎回基金
- **shares** (float) -- 赎回份额

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 赎回200份'华夏成长混合'(000001)
order = redeem_shares('000001', 200)
```

redeem_value - 根据赎回总金额计算份额[🔗](#)

deepquant.quest.mod.mod_fund.api.redeem_value(*market_code*, *cash_amount*)[🔗](#)

根据赎回总金额计算份额

Parameters:

- **market_code** (Union[str, [Instrument](#)]) -- 赎回基金
- **cash_amount** (float) -- 赎回金额

Return type:

Optional[[Order](#)]

Returns:

None or Order

Example:

```
# 赎回价值5000块'华夏成长混合'(000001)
order = redeem_value('000001', 5000)
```

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