---

scheduler 定时器¶

scheduler.run_daily - 每天运行¶

scheduler.run_daily(*function*)¶

每日运行一次指定的函数，只能在init内使用。

注意，schedule一定在其对应时间点的handle_bar之后执行。

Parameters:

function (*func*) -- 使传入的function每日运行。注意，function函数一定要包含（并且只能包含）context, bar_dict两个输入参数

Example:

以下的范例代码片段是一个非常简单的例子，在每天交易后查询现在portfolio中剩下的cash的情况:

```
#scheduler调用的函数需要包括context, bar_dict两个输入参数
def log_cash(context, bar_dict):
    logger.info("Remaning cash: %r" % context.portfolio.cash)

def init(context):
    #...
    # 每天运行一次
    scheduler.run_daily(log_cash)
```

scheduler.run_weekly - 每周运行¶

scheduler.run_weekly(*function*, *weekday=x*, *tradingday=t*)¶

每周运行一次指定的函数，只能在init内使用。

注意:

- tradingday中的负数表示倒数。
- tradingday表示交易日。如某周只有四个交易日，则此周的tradingday=4与tradingday=-1表示同一天。
- weekday和tradingday不能同时使用。

Parameters:

- **function** (*func*) -- 使传入的function每日交易开始前运行。注意，function函数一定要包含（并且只能包含）context, bar_dict两个输入参数。

- **weekday** (*int*) -- 1~5 分别代表周一至周五，用户必须指定
- **tradingday** (*int*) -- 范围为[-5,1],[1,5] 例如，1代表每周第一个交易日，-1代表每周倒数第一个交易日，用户可以不填写。

Example:

以下的代码片段非常简单，在每周二固定运行打印一下现在的portfolio剩余的资金:

```
#scheduler调用的函数需要包括context, bar_dict两个参数
def log_cash(context, bar_dict):
    logger.info("Remaning cash: %r" % context.portfolio.cash)

def init(context):
    #...
    # 每周二打印一下剩余资金：
    scheduler.run_weekly(log_cash, weekday=2)
    # 每周第二个交易日打印剩余资金：
    #scheduler.run_weekly(log_cash, tradingday=2)
```

scheduler.run_monthly - 每月运行¶

scheduler.run_monthly(*function*, *tradingday=t*)¶

每月运行一次指定的函数，只能在init内使用。

注意:

- tradingday的负数表示倒数。
- tradingday表示交易日，如某月只有三个交易日，则此月的tradingday=3与tradingday=-1表示同一。

Parameters:

- **function** (*func*) -- 使传入的function每日交易开始前运行。注意，function函数一定要包含（并且只能包含）context, bar_dict 两个输入参数。
- **tradingday** (*int*) -- 范围为[-23,1], [1,23] ，例如，1代表每月第一个交易日，-1代表每月倒数第一个交易日，用户必须指定。

Example:

以下的代码片段非常简单的展示了每个月第一个交易日的时候我们进行一次财务数据查询，这对根据财务数据来调节股票组合的策略会非常有用:

```
#scheduler调用的函数需要包括context, bar_dict两个参数
def query_fundamental(context, bar_dict):
        # 查询revenue前十名的公司的股票并且他们的pe_ratio在25和30之间。打fundamentals的时候会有auto-complete方便写查询代码。
    fundamental_df = get_fundamentals(
        query(
            fundamentals.income_statement.revenue, fundamentals.eod_derivative_indicator.pe_ratio
        ).filter(
            fundamentals.eod_derivative_indicator.pe_ratio > 25
        ).filter(
            fundamentals.eod_derivative_indicator.pe_ratio < 30
        ).order_by(
            fundamentals.income_statement.revenue.desc()
        ).limit(
            10
        )
    )

    # 将查询结果dataframe的fundamental_df存放在context里面以备后面只需：
    context.fundamental_df = fundamental_df

    # 实时打印日志看下查询结果，会有我们精心处理的数据表格显示：
    logger.info(context.fundamental_df)
    update_universe(context.fundamental_df.columns.values)

 # 在这个方法中编写任何的初始化逻辑。context对象将会在你的算法策略的任何方法之间做传递。
def init(context):
    # 每月的第一个交易日查询以下财务数据，以确保可以拿到最新更新的财务数据信息用来调整仓位
    scheduler.run_monthly(query_fundamental, tradingday=1)
```

time_rule - 定时间运行¶

scheduler还可以用来做定时间运行，比如在每天开盘后的一小时后或一分钟后定时运行，这里有很多种组合可以让您达到各种自己想要达到的定时运行的目的。

使用的方法是和上面的 scheduler.run_daily() , scheduler.run_weekly() 和 scheduler.run_monthly() 进行组合加入time_rule来一起使用。

注意:

- market_open与market_close都跟随中国A股交易时间进行设置，即09:31~15:00。
- physical_time用于设置物理时间，与market_open和market_close的相对时间不同。

- physical_time更多用于交易时间不确定的品种，如商品期货。
- 使用time_rule定时运行会在分钟级别回测和实时模拟交易中有定义的效果，在日回测中只会默认依然在该天运行，并不能在固定的时间运行。
- 在分钟回测中如未指定time_rule,则默认在开盘后一分钟运行,即09:31分。
- 目前暂不支持开盘交易(即 09:30分交易)。
- market_open(minute=120)将在11:30执行， market_open(minute=121)在13:01执行，中午休市的区间会被忽略。
- time_rule='before_trading'表示在开市交易前运行scheduler函数。该函数运行时间将在before_trading函数运行完毕之后handle_bar运行之前。

time_rule: 定时具体几点几分运行某个函数。time_rule='before_trading' 表示开始交易前运行；market_open(hour=x, minute=y)表示A股市场开市后x小时y分钟运行，market_close(hour=x, minute=y)表示A股市场收市前x小时y分钟运行。如果不设置time_rule默认的值是中国A股市场开市后一分钟运行。

market_open, market_close, physical_time参数如下：

参数	类型	注释
hour	int - option [1,4]	具体在market_open/market_close后/前第多少小时执行, 股票的交易时间为[9:31 - 11:30],[13:01 - 15:00]共240分钟，所以hour的范围为 [1,4]
minute	int - option [1,240]	具体在market_open/market_close的后/前第多少分钟执行,同上，股票每天交易时间240分钟，所以minute的范围为 [1,240],中午休市的时间区间会被忽略。

example:

- 每天的开市后10分钟运行:

```
1 scheduler.run_daily(function, time_rule=market_open(minute=10))
```

- 每周的第t个交易日闭市前1小时运行:

```
1 scheduler.run_weekly(function, tradingday=t, time_rule=market_close(hour=1))
```

- 每月的第t个交易日开市后1小时运行:

```
1 scheduler.run_monthly(function, tradingday=t, time_rule=market_open(hour=1))
```

- 每天开始交易前运行:

```
1 scheduler.run_daily(function, time_rule='before_trading')
```

- 每天十点运行

```
1 scheduler.run_daily(function, time_rule=physical_time(hour=10, minute=0))
```

Next Previous