**标的相关数据**

**一、标的相关信息查询**

|  |
| --- |
| 【相关信息】：主要包含标的的交易所、类型、交易时间、支持的数据粒度、上市及退市时间等，标的自身的属性信息。 |

**调用方法**

|  |
| --- |
| Python context.get\_symbol\_information(name=None) |

**方法说明**

|  |
| --- |
| 根据**标的代码**，查询其相关的**属性信息。**  只能在choose\_stock函数内部调用，不能单独使用，该函数在每根k线都会执行一次。  由于基础信息一般不会有太大变化，建议只在策略开始时取一次即可。 |

**参数说明**

* name：标的唯一的标识代码。需包含交易所，对于没有交易所的场外基金，对应位置设为“NONE”，例如："600000.XSHG" 或 "000003.NONE"

**代码示例**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["399804.XSHE"]    if context.count == 0:  data = context.get\_symbol\_information("399804.XSHE")  context.log(data) |

**返回数据**

|  |
| --- |
| JSON {  'symbol': '399804',   'timezone': 'Asia/Shanghai',   'minmov': 1,   'exchange': 'XSHE',   'juejin\_exchange': 'SZSE',   'juejin\_symbol': '399804',   'session': '0930-1130,1300-1500',   'has\_intraday': True,   'has\_no\_volume': False,   'has\_daily': True,   'description': '中证体育',   'type': 'index',   'data\_status': 'endofday',   'supported\_resolutions': ['1S', '1', '1D', '2D', '3D', '1W', '3W', '1M', '6M'],   'has\_seconds': True,   'has\_weekly\_and\_monthly': True,   'pricescale': 10000,   'start\_date': '2015-07-28',   'end\_date': '2021-12-02 16:37:37',   'expired': False,   'minute\_valid\_start': 1451577600000,   'tick\_valid\_start': 1630658257000,   'volume\_precision': 2 } |

**字段说明**



**点击图片可查看完整电子表格**

**二、指数成分股查询**

**调用方法**

|  |
| --- |
| Python context.get\_index\_stocks(  symbol\_exchange,  date=None,  weight\_mode=False ) |

**方法说明**

|  |
| --- |
| 根据指数的标的代码，指定的查询日期，及权重模式，查询该指数的成分股。  只能在choose\_stock函数内部调用，不能单独使用，该函数在每根k线都会执行一次。 |

**参数说明**

* symbol\_exchange：str，标的唯一的标识代码
* date：str，查询日期
* weight\_mode：bool，是否需要返回权重参数，默认不需要

**代码示例1**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["600000.XSHG"]    if context.count == 0:  data = context.get\_index\_stocks("000016.XSHG", date="2021-09-30")  context.log(data)  context.count += 1 |

**返回数据**

|  |
| --- |
| JSON ['601628.XSHG', '600016.XSHG', ...，'600028.XSHG'] |

**代码示例2**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["600000.XSHG"]    if context.count == 0:  data = context.get\_index\_stocks("000016.XSHG", date="2021-09-30", weight\_mode=True)  context.log(data)  context.count += 1 |

**返回数据**

|  |
| --- |
| JSON [  {'600276.XSHG': 3.0199999809265137},   {'601888.XSHG': 3.4100000858306885},   ...，  {'601066.XSHG': 0.4000000059604645} ] |

**三、标的所存在的所有指数查询**

**调用方法**

|  |
| --- |
| Python context.get\_symbol\_exist\_indexs(symbol\_exchange, date) |

**方法说明**

|  |
| --- |
| 根据标的唯一标识代码和查询日期，获取该标的所在的指数列表，具体为不超过指定日期的最近一天更新的结果。  只能在choose\_stock函数内部调用，不能单独使用，该函数在每根k线都会执行一次。 |

**参数说明**

* symbol\_exchange：str，标的唯一的标识代码
* date：str，查询日期

**代码示例**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["600000.XSHG"]    if context.count == 0:  data = context.get\_symbol\_exist\_indexs("600010.XSHG", "2021-10-31")  context.log(data)  context.count += 1 |

**返回数据**

|  |
| --- |
| JSON ['399317.XSHE', '000902.XSHG', ..., '000123.XSHG'] |

**四、标的名称是否带有ST或\*ST查询**

|  |
| --- |
| 【ST】：公司经营连续二年亏损，特别处理  【\*ST】：公司经营连续三年亏损，退市预警； |

**调用方法**

|  |
| --- |
| Python context.judge\_st(symbol\_exchange) |

**方法说明**

|  |
| --- |
| 根据标的唯一标识代码，获取该标的名称中是否带有ST或\*ST。  只能在choose\_stock函数内部调用，不能单独使用，该函数在每根k线都会执行一次。 |

**参数说明**

* symbol\_exchange：str，标的唯一的标识代码

**代码示例**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["600000.XSHG"]    if context.count == 0:  data = context.judge\_st("000001.XSHE")  context.log(data)  context.count += 1 |

**返回数据**

|  |
| --- |
| JSON False |

**五、非法标的查询**

|  |
| --- |
| 【非法标的】：不存在的标的代码 |

**调用方法**

|  |
| --- |
| Python context.judge\_invalid\_symbol(symbol\_exchange) |

**方法说明**

|  |
| --- |
| 根据输入的标的唯一标识代码，判断各标的是否在当前的合法标的池中，返回非法标的。  只能在choose\_stock函数内部调用，不能单独使用，该函数在每根k线都会执行一次。 |

**参数说明**

* symbol\_exchange：list，待查询的所有标的标识代码

**代码示例**

|  |
| --- |
| Python def indicators(context):  context.count = 0  def choose\_stock(context):  context.symbol\_list = ["600000.XSHG"]    if context.count == 0:  data = context.judge\_invalid\_symbol(["000001", "abcdes", "600000.XSHG", "600000.XSHE"])  context.log(data)  context.count += 1 |

**返回数据**

|  |
| --- |
| JSON ['abcdes', '600000.XSHE'] |