



Choice 数据

Python接口使用手册

V2.3.3.0

- 电话：400-620-1818 邮箱：Choiceinfo@eastmoney.com
- 接口官网：<http://quantapi.eastmoney.com/>
- 地址：上海市徐汇区龙田路190号2号楼东方财富大厦

版本号	2.3.3.0	发布时间	2018/12/27
版本变动信息: V2.3.2.0 -> V2.3.3.0			
<ol style="list-style-type: none">1. css和csd函数新支持组合指标2. preport函数新增查询组合业绩贡献-已清仓股票、周期回报-阶段回报、盈亏分析、风险分析、VAR分析、组合日报、交易日报报表的功能3. csq函数新增date字段，返回数据下发时的日期4. cmc函数支持期限由7天改为30天5. css/csd新增代码校验功能6. css截面函数新增指标共509个，csd序列函数新增指标共15个，ctr报表函数新增指标共1个，详见 http://choiceclub.jg.eastmoney.com/?/article/53837. 部分指标精度调整，详见安装包内readme文件8. 其他bug修复			

目录

1. PYTHON 接口配置	5
1.1 文件存放位置	5
1.2 EMQUANTAPI PYTHON 接口配置的系统环境要求与方法	5
1.2.1 系统环境要求	5
1.2.2 配置方法	6
2. PYTHON 应用示例	7
2.1 示例项目存放位置	7
2.2 示例项目使用	7
3. 接口函数	7
3.1 接口功能函数列表	7
3.2 接口功能函数明细	9
3.2.1 登录函数 : start	9
3.2.2 退出函数 : stop	10
3.2.3 序列函数 : csd	11
3.2.4 截面函数 : css	15
3.2.5 历史分钟 k 线 : cmc	17
3.2.6 报价订阅函数 : csq	19
3.2.7 快照函数 : csqsnapshot	21
3.2.8 日内跳价函数 : cst	23
3.2.9 专题报表函数 : ctr	24
3.2.10 条件选股函数 : cps	26

3.2.11 宏观数据函数 : edb.....	31
3.2.12 宏观指标信息查询函数 : edbquery.....	33
3.2.13 新建组合函数 : pcreate	36
3.2.14 组合账户关系查询函数 : pquery.....	38
3.2.15 组合批量下单函数 : porder.....	41
3.2.16 组合报表查询函数 : preport.....	43
3.2.17 删除组合函数 : pdelete.....	45
3.2.18 取消报价订阅 : csqcancel	46
3.2.19 板块函数 : sector.....	47
3.2.20 交易日函数 : tradedates	48
3.2.21 偏移 N 天的交易日函数 : getdate.....	51
3.2.22 区间交易日天数函数 : tradedatesnum	53
3.2.23 设置代理函数 : setproxy	55
3.2.24 配置函数 : setserverlistdir.....	56
4. 错误类型定义列表	56

1. Python 接口配置

1.1 文件存放位置

在量化接口官网 (<http://quantapi.eastmoney.com/>) 下载压缩包

EmQuantAPI_Python.RAR, 下载完成后解压。其中：

installEmQuantAPI.py 用于注册 python 接口

libs 含接口的 lib 和 dll，以及接口登录激活工具 LoginActivator.exe 文件，通过手机号获取验证码，自动生成令牌用以登录；

demo.py 为示例脚本文件

1.2 EMQuantAPI Python 接口配置的系统环境要求与方法

1.2.1 系统环境要求

- Windows 系统，支持 32 位和 64 位系统；
- Python 2.6.x, 2.7.x, 3.x
- Microsoft Visual C++ 2010 Redistributable Package(可在电脑“程序和功能”中查看是否安装)
 - 32 位：<https://www.microsoft.com/zh-cn/download/confirmation.aspx?id=5555>
 - 64 位：<https://www.microsoft.com/zh-CN/download/details.aspx?id=14632>

1.2.2 配置方法

- 在命令行运行 `installEmQuantAPI.py` 注册，若使用 Anaconda 编译器，请在编译器中运行 `installEmQuantAPI.py` 注册。
- 绑定手机号

登录东方财富主页 (<http://www.eastmoney.com/>)，进入用户中心-个人设置-账号绑定 (<https://passport2.eastmoney.com/pub/passsafe>) 绑定手机号。

或者登录 Choice 金融终端，进入用户中心-资料管理绑定手机号。

- 登录激活

运行接口激活工具 `LoginActivator.exe`，输入绑定手机号获取验证码，激活成功后生成令牌文件 `userInfo`，用户使用时无需输入用户名和密码，默认从令牌中获取登陆权限。一个账号最多支持在十台设备上激活。



2. Python 应用示例

2.1 示例项目存放位置

demo.py

2.2 示例项目使用

在 Python GUI 中按照 demo.py 的方式调用

EmQuantAPI.py 文件中有一个 DemoCallback 函数，为报价订阅回调函数的实例，可参考使用

EmQuantAPI.py 文件中有一个 cstCallBack 函数，为日内跳价回调函数的实例，可参考使用

3. 接口函数

3.1 接口功能函数列表

登录函数	start	接口登录函数
退出函数	stop	接口退出函数
数据函数	csd	获取序列数据(同步)
	css	获取截面数据(同步)
	cmc	获取历史分钟 K 线数据(同步)
	csq	报价订阅函数(异步)

	csqsnapshot	快照函数(同步)
	cst	日内跳价函数(异步)
	ctr	专题报表函数(同步)
	cps	条件选股函数 (同步)
	edb	宏观数据函数(同步)
	edbquery	宏观指标信息查询函数(同步)
	pcreate	新建组合(同步)
	pquery	组合账户关系查询函数(同步)
	porder	组合批量下单函数(同步)
	preport	组合报表查询(同步)
	pdelete	删除组合(同步)
功能函数	csqcancel	取消报价订阅(异步)
	sector	获取系统板块成分(同步)
	tradedates	获取区间内交易日(同步)
	gettradedate	获取偏移 N 天的交易日(同步)
	tradedatesnum	区间交易日天数函数(同步)
工具函数	setproxy	设置网络代理函数

配置函数	setserverlistdir	设置 ServerList.json.e 文件存放目录
------	------------------	-----------------------------

3.2 接口功能函数明细

3.2.1 登录函数：start

EQErr start(options, logcallback, mainCallBack);				
初始化登录函数，登录验证通过以后，即可正常使用接口函数获取数据，用户名和密码传空				
参数名	参数简称	定义		描述
options	可选参数	字符串	输入	附加参数，可填附加字段。见 附注 1
logcallback	可选参数	c_LogCallback	输入	日志回调函数
mainCallBack	可选参数	c_DataCallback	输入	主回调函数
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>			
范例 (Python2.x)	<pre>from EmQuantAPI import * loginresult = c.start() #loginresult为c.EmQuantData类型数据 print loginresult</pre>			

<p>范例</p> <p>(Python3.x)</p>	<pre>from EmQuantAPI import * loginresult = c.start() #loginresult为c.EmQuantData类型数据 print (loginresult)</pre>
--------------------------------------------	-------------------------------------------------------------------------------------------------------------------

附注 1 登录函数可选参数列表：

中文名称	英文名称	取值范围	说明
服务器测速	TestLatency	取值范围 0,1	取值 0，不测速，连接默认服务器
		缺省值：0	取值 1，登录前服务器测速，并保存为默认
强制登录	ForceLogin	取值范围 0,1	取值 0，当线上已存在该账户时，不强制登录
		缺省值：0	取值 1，当线上已存在该账户时，强制登录，将前一位在线用户踢下线
记录登录信息标记	RecordLoginInfo	取值范围 0,1	取值 0，不记录
		缺省值：1	取值 1，追加记录登录信息到 logininfo.log 文件，文件位于 serverlist.json.e 所在目录下

3.2.2 退出函数：stop

stop()				
退出登录				
参数名	参数简称	定义		描述
无				
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>			

<p>范例 (Python2.x)</p>	<pre>from EmQuantAPI import * loginresult = c.start() print loginresult #do something... logoutresult = c.stop() #logoutresult为c.EmQuantData类型数据 print logoutresult</pre>
<p>范例 (Python3.x)</p>	<pre>from EmQuantAPI import * loginresult = c.start() print (loginresult) #do something... logoutresult = c.stop() #logoutresult为c.EmQuantData类型数据 print (logoutresult)</pre>

3.2.3 序列函数：csd

<p>csd(codes, indicators, startdate=None, enddate=None, options=None, *arga, **argb)</p>				
<p>获取股票，指数，基金，期货，外汇等各个证券品种或组合的日频历史行情序列数据</p>				
参数名	参数简称	定义		描述

codes	证券代码	字符串或者序列	输入	东财代码，支持多代码输入，以半角逗号分隔， 不支持跨品种证券输入
indicators	指标简称	字符串或者序列	输入	指标名称，支持多指标输入，以半角逗号分隔， 最多不超过 64 个 ，详细指标列表见 指标手册
startdate	起始日期	字符串或者 datetime	输入	支持格式： YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
EndDate	截止日期	字符串或者 datetime	输入	支持格式： YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
options	可选参数	字符串	输入	附加参数，可填附加字段，见 附注 2
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体 <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 </pre>			

	<pre> self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果 </pre>
<p>范例</p> <p>(Python2.x)</p>	<pre> data = c.csd("300059.SZ,600425.SH", "open,close", "2016-07-01", "2016-07-06") if data.ErrorCode != 0: print "request csd Error, ", data.ErrorMsg else: for code in data.Codes: for i in range(0, len(data.Indicators)): for j in range(0, len(data.Dates)): print data.Data[code][i][j] </pre>
<p>范例</p> <p>(Python3.x)</p>	<pre> data = c.csd("300059.SZ,600425.SH", "open,close", "2016-07-01", "2016-07-06") if data.ErrorCode != 0: print("request csd Error, ", data.ErrorMsg) else: for code in data.Codes: for i in range(0, len(data.Indicators)): for j in range(0, len(data.Dates)): print(data.Data[code][i][j]) </pre>

附注 2 序列函数可选参数列表：

中文名称	英文名称	取值范围	说明
是否输出 pandas 格式	IsPandas	0--1, 缺省值: 0	非 pandas 格式--0; pandas 格式--1; 需要安装 pandas 包;
pandas 索引	RowIndex	1--2, 缺省值: 1	证券代码--1; 日期--2;
日期周期	Period	1--4, 缺省值: 1	日期周期: 日, 周, 月, 年 分别对应: 1, 2, 3, 4
复权方式	AdjustFlag	1--3, 缺省值: 1	不复权--1; 后复权--2; 前复权--3.
币种	CurType	1--4, 缺省值: 1	原始币种--1; 人民币--2; 美元--3; 港币--4.
债券价格类型	PriceType	1--2, 缺省值: 1	净价--1; 全价--2.
按日期排序	Order	1--2, 缺省值: 1	升序--1; 降序--2
市场类型	Market	见说明, 缺省值: "CNSESH"	CNSESH 上海证券交易所 CNSESZ 深圳证券交易所 HKSE00 香港证券交易所 USSE00 美国证券交易所 USSEND 美国纳斯达克市场 USSENY 纽约证券交易所 CNIBBD 中国银行间债券市场 CNFEBC 渤海商品交易所 CNFEDC 大连商品交易所 CNFESF 上海期货交易所 CNFEZC 郑州商品交易所 CNFF00 中国金融期货交易所 0 自然日 CNSH00 沪股通交易日 CNSHHK 沪港股通交易日 CNSZ00 深股通交易日 CNSZHK 深港股通交易日
空值替换	ShowBlank	整数	对返回数据中的空值的进行特殊处理, 例如: ShowBlank=0, 所有的空值都替换成 0

3.2.4 截面函数：css

css(codes, indicators, options=None, *arga, **argb)				
获取股票，指数，基金，期货，外汇，组合等各个证券品种或组合的基本资料，财务，估值等截面数据				
参数名	简称	定义		描述
codes	证券代码	字符串或者序列	输入	东财代码，支持多代码输入，以半角逗号分隔， 不支持跨品种证券输入
indicators	指标简称	字符串或者序列	输入	指标名称，支持多指标输入，以半角逗号分隔， 最多不超过 64 个 ，详细指标列表见 指标手册
options	可选参数	字符串	输入	附加参数，可填附加字段，见 附注 3
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体 <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列（返回当前的系统日期） self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 </pre>			

	self.Data = dict() #数据结果
范例 (Python2.x)	<pre> data = c.css("300059.SZ", "open,close") if data.ErrorCode != 0: print "request css Error, ", data.ErrorMsg else: for code in data.Codes: for i in range(0, len(data.Indicators)): print data.Data[code][i] </pre>
范例 (Python3.x)	<pre> data = c.css("300059.SZ", "open,close") if data.ErrorCode != 0: print("request css Error, ", data.ErrorMsg) else: for code in data.Codes: for i in range(0, len(data.Indicators)): print(data.Data[code][i]) </pre>

附注 3 截面函数可选参数列表：

中文名称	英文名称	取值范围	说明
是否输出 pandas 格式	ispandas	0--1，缺省值：0	非 pandas 格式--0； pandas 格式--1； 需要安装 pandas 包；

pandas 索引	RowIndex	1--2, 缺省值: 1	证券代码--1; 日期--2;
空值替换	ShowBlank	整数	对返回数据中的空值的进行特殊处理, 例如: ShowBlank=0, 所有的空值都替换成 0

(注: 截面函数每分钟请求次数不能超过 700 次)

3.2.5 历史分钟 k 线: cmc

cmc(codes, indicators, startdate=None, enddate=None, options=None, *arga, **argb)				
获取最近30个自然日沪深股票以及大连期货交易所、上海期货交易所、郑州商品期货交易所、上海国际能源交易中心期货的历史分钟K线序列数据 (商品期货目前只支持一分钟级别的历史数据)				
参数名	参数简称	定义		描述
codes	证券代码	字符串或者序列	输入	东财代码, 只支持单个代码, 支持沪深股票和商品期货
indicators	指标简称	字符串或者序列	输入	指标名称, 支持多指标输入, 以半角逗号分隔, 详细指标列表见 指标手册
startdate	起始日期	字符串或者 datetime	输入	支持格式: YYYYMMDD, YYYY/MM/DD YYYY/M/D, YYYY-MM-DD, YYYY-M-D
EndDate	截止日期	字符串或者 datetime	输入	支持格式:

				YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
options	可选参数	字符串	输入	附加参数, 可填附加字段, 见 附注 4
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果</pre>			
范例 (Python2 .x)	<pre>data = c.cmc("300059.SZ", "OPEN,CLOSE,HIGH", "2018-03-10", "2018-03-11") for i in range(0, len(data.Indicators)): for j in range(0, len(data.Dates)): print "indicator=%s, value=%s" % (data.Indicators[i], str(data.Data[i][j]))</pre>			

范例 (Python3.x)	<pre>data = c.cmc("300059.SZ", "OPEN,CLOSE,HIGH", "2018-03-10", "2018-03-11") for i in range(0, len(data.Indicators)): for j in range(0, len(data.Dates)): print("indicator=%s, value=%s" % (data.Indicators[i], str(data.Data[i][j])))</pre>
-------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

附注 4 历史分钟 K 线函数函数可选参数列表：

中文名称	英文名称	取值范围	说明
是否输出 pandas 格式	IsPandas	0--1, 缺省值：0	非 pandas 格式--0 ; pandas 格式--1 ; 需要安装 pandas 包;
pandas 索引	RowIndex	1--2, 缺省值：1	证券代码--1 ; 日期和时间--2 ;
日期周期	Period	正整数, 缺省值：1	单位, 分钟, 取值为 1 至 240 (仅支持沪深股票修改日期周期)
复权方式	AdjustFlag	1, 2, 3, 缺省值：1	不复权--1 ; 后复权--2 ; 前复权--3 ;
是否包含历史数据	IsHistory	0,1, 缺省值：1	历史分钟--1 ; 当日分钟--0
复权基期	BaseDate	取值>=截止日期, 缺省值：当日日期	支持 YYYYMMDD,YYYY-MM-DD,YYYY/MM/DD, 例：20160101

(注：交易所盘后会推送修正数据，当日分钟最后一笔数据根据最新修正数据变动)

3.2.6 报价订阅函数：csq

csq (codes, indicators, options=None, fncallback=None, userparams=None, *arga, **argb)
提供沪深，美股，指数，基金，债券，期货，外汇，利率等各个证券品种的报价及买卖档数据订阅

参数名	简称	定义		描述
codes	证券代码	字符串或者序列	输入	东财代码，支持多代码输入，以半角逗号分隔
indicators	指标简称	字符串或者序列	输入	指标名称，支持多指标输入，以半角逗号分隔，详细指标列表见 指标手册 ，其中沪深股票的行情报价、资金流向、DDE 三类指标需在 csq 语句中分开使用
options	可选参数	字符串	输入	附加参数，可填附加字段，见 附注 5
fncallback	回调函数	c_DataCallback	输入	报价订阅回调
userparams		字符串	输入	用户参数，回调时原样返回
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果</pre>			
范例	<pre>data = c.csq("002716.SZ", "PRECLOSE,OPEN,HIGH,LOW,NOW,AMOUNT")</pre>			

--	--

附注 5 报价订阅函数可选参数列表：

中文名称	英文名称	取值范围	说明
推送方式	Pushtype	取值范围 0,1 缺省值：0	取值 0，增量推送 取值 1，全量推送
待处理队列堆积警告数量	WarnSize	默认值 500	当待处理队列达到设置数量时会有一条日志输出，若队列数一直大于设置数量，则每 5 秒输出一次日志。本参数在初次调用 csq 时生效。

注：关于推送方式的说明

增量推送指的是当订阅代码的指标数据有变动时，变动指标推送对应数值，无变动时，推送 none；

全量推送指的是当订阅代码的指标数据有变动时，全部代码和指标数据都会做推送，可能存在大量重复数据

3.2.7 快照函数：csqsnapshot

csqsnapshot(codes, indicators, options="")				
提供沪深，美股，指数，基金，债券，期货，外汇，利率等各个证券品种的报价及买卖档数据快照				
参数	简称	定义		描述
codes	证券代码	字符串或者序列	输入	东财代码，支持多代码输入，以半角逗号分隔

indicators	指标简称	字符串或者序列	输入	指标名称，支持多指标输入，以半角逗号分隔，详细指标列表见 指标手册
options	可选参数	字符串	输入	附加参数，可填附加字段，见 附注 6
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果</pre>			
范例 (Python2.x)	<pre>data = c.csqsnapshot("000005.SZ", "PRECLOSE,OPEN,HIGH,LOW") if data.ErrorCode != 0: print "request ctr Error, ", data.ErrorMsg else: for key,value in data.Data.items(): for v in value: print v, print ""</pre>			
范例 (Python3.x)	<pre>data = c.csqsnapshot("000005.SZ", "PRECLOSE,OPEN,HIGH,LOW") for key,value in data.Data.items(): print(key, ">>> ", end="")</pre>			

	<pre> for v in value: print(v, " ", end="") print()</pre>
--	-----------------------------------------------------------------

附注 6 快照函数可选参数列表：

中文名称	英文名称	取值范围	说明
是否输出 pandas 格式	ispandas	0--1, 缺省值: 0	非 pandas 格式--0; pandas 格式--1; 需要安装 pandas 包;

（注：快照函数请求频次不能超过每 3 秒/次）

3.2.8 日内跳价函数：cst

cst(codes, indicators, startdatetime, enddatetime, options = None, fncallback=None, userparams=None)				
提供沪深股票以及大连期货交易所、上海期货交易所、郑州商品期货交易所期货的当日跳价				
参数名	简称	定义		描述
codes	证券代码	字符串或者序列	输入	东财代码，支持多代码输入，以半角逗号分隔， 不支持跨品种证券输入
indicators	指标简称	字符串或者序列	输入	指标名称，支持多指标输入，以半角逗号分隔，详细指标列表见 指标手册
startdatetime	开始时间	字符串	输入	支持格式 YYYYMMDDHHMMSS 或 HHMMSS

enddatetime	结束时间	字符串	输入	支持格式 YYYYMMDDHHMMSS 或 HHMMSS
options	可选参数	字符串	输入	预留
fncallback	回调函数	c_DataCallback	输入	日内跳价回调
userparams		字符串	输入	用户参数，回调时原样返回
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果</pre>			
范例	<pre>data = c.cst('300059.SZ,600000.SH', 'TIME,OPEN,HIGH,LOW,NOW', '100000', '101000')</pre>			

3.2.9 专题报表函数：ctr

ctr(ctrName, indicators="", options="")
提供专题报表数据

参数	简称	定义		描述
ctrName	报表名称	字符串	输入	东财报表名称，详细枚举见 指标手册
indicators	报表字段简称	字符串或者 序列	输入	报表字段简称，支持多字段输入，以半角逗号分隔，传空或匹配不到时展示报表全部字段，字段枚举详见 指标手册
options	报表参数	字符串	输入	报表参数明细，详见 指标手册
返回值	<p>EmQuantData 为结构体</p> <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果 </pre>			
范例 (Python2.x)	<pre> data = c.ctr("INDEXCOMPOSITION", "", "IndexCode=000001.SH,EndDate=2017-01-13") if data.ErrorCode != 0: print "request ctr Error, ", data.ErrorMsg else: for key,value in data.Data.items(): for v in value: </pre>			

	<pre>print v, print ""</pre>
范例 (Python3.x)	<pre>data = c.ctr("INDEXCOMPOSITION", "", "IndexCode=000001.SH,EndDate=2017-01-13") for key,value in data.Data.items(): for v in value: print(v, " ", end=" ") print()</pre>

3.2.10 条件选股函数：cps

cps(cpsCodes, cpsIndicators, cpsConditions, cpsOptions)				
条件选股函数				
参数	简称	定义		描述
cpsCodes	板块代码 或证券代 码	字符串、列表、 元组	输入	<p>控制选股范围，CPS 函数只能选取沪深的板 块和证券代码，取值格式有两种：</p> <p>1. 板块代码，以 B_开头，如 "B_001004"，常见板块代码见附注 7；</p> <p>2. 东财代码，多个代码间用半角逗号隔开， 如"000001.SZ,000002.SZ,600000.SH"</p>

cpsIndicators	条件参数	字符串	输入	定义条件表达式使用的参数，多个参数之间用英文分号隔开,内部各参数用半角逗号隔开，具体指标和英文简称见 指标手册 ，如： s1,open,2016/12/31,1;s2,close,2017/02/25,1;s3,NAME;s4, LISTDATE
cpsConditions	条件表达式	字符串	输入	条件表达式，各表达式用 and 连接，表达式支持的操作符：ANY，CONTAINALL，ISNULL，ISNOTNULL，比较运算符，算术运算符，逻辑运算符(必须小写)如 and、or、not 等，具体操作符释义详见 附注 8 ； 条件参数引用格式：[参数名 1]，例如： [s1] >10 and [s2] > [s1] and not CONTAINANY ([s3],重工,银行)；若选择的条件是日期，需加 d()，例如： [s4]>d(2017/7/21)，若选日期区间，需用多项日期表达式，用 and 连接，例如： [s4]>d(2013-09-30) and [s4]<d(2014-07-10)
cpsOptions	附加参数	字符串	输入	其他附加条件，如排序、取前 N 条选股结果等，具体使用规则见 附注 9
返回值	EmQuantData 为结构体			

	<pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果 </pre>
<p>范例 (Python2.x)</p>	<pre> data = c.cps("B_001004", " s1,OPEN,2017/2/27,1; s2,NAME", "[s1]>0", " orderby=rd([s1]),top=max([s1],100)") if data.ErrorCode != 0: print "request cps Error, ", data.ErrorMsg else: for it in data.Data: print it </pre>
<p>范例 (Python3.x)</p>	<pre> data = c.cps("B_001004", " s1,OPEN,2017/2/27,1; s2,NAME", "[s1]>0", " orderby=rd([s1]),top=max([s1],100)") print("cps 输出结果=====分割线=====") for it in data.Data: print(it) </pre>

附注 7 常见板块代码：

所属领域	板块名称	板块代码
沪深股票	全部 A 股	001004
沪深股票	上证 A 股	001005
沪深股票	深证 A 股	001006
沪深股票	深证 B 股	001013
沪深股票	上证 B 股	001012
沪深股票	全部 B 股	001011
沪深股票	创业板	001010
沪深股票	中小板	001009
沪深股票	深证主板	001008
沪深股票	深证主板 A 股	001007
沪深股票	风险警示股票	001023
沪深股票	风险警示股票（深交所）	001025
沪深股票	风险警示股票（上交所）	001024
沪深股票	已发行待上市股票	001020
沪深股票	正在发行的股票	001019
沪深股票	*ST	001018
沪深股票	ST	001017
沪深股票	全部 A 股(非金融石油石化)	001044
沪深股票	可转债标的	001046
沪深股票	融资融券标的	001045
沪深股票	深股通	001041
沪深股票	沪深股通	001047
沪深股票	深证主板 B 股	001033
沪深股票	中小板(含 ST,ST*)	001032
沪深股票	深证主板 A 股(含 ST,ST*)	001031
沪深股票	沪股通	001038
沪深股票	中证 500 成份	009006062
沪深股票	中证 1000 成份	009007552
沪深股票	上证 50 指数成份	009007063
沪深股票	上证 180 指数成份	009007060
沪深股票	创业板综成份	009007145
沪深股票	创业板指成份	009007144
沪深股票	中小板综成份	009007125
沪深股票	中小板指成份	009007124
沪深股票	上证综合指数成份	009007104
沪深股票	沪深 300 成份	009006195
沪深股票	深证综合指数成份	009007251
沪深股票	MSCI 中国（概念类）	007230
沪深股票	预盈预增	007054
沪深股票	预亏预减	007053

附注 8 操作符列表：

分类	操作符	描述	详细	举例
算术运算符	+ - * /	加 减 乘 除		
比较运算符	> = < >= <= <>	大于 等于 小于 大于或等于 小于或等于 不等于		
逻辑运算符	and or not	与 或 非		
	ANY CONTAINANY	包含任意一个		ANY(s[1], 中国, 美国) 表示当变量 s1 中包含 “中国” 或 “美国” 则成立
	CONTAINALL	包含所有值		CONTAINALL (s[1], 中国, 美国) 表示当变量 s1 中包含 “中国” 且包含 “美国” 则成立
	MAX	取最大的 N 个值	用在 Top 表达式中, 对选股结果取 TOP	top=max([s1],100)
	MIN	取最小的 N 个值	用在在 Top 表达式中, 对选股结果取最小的 N 行	top=min([s2],100)
	ISNULL ISNOTNULL	等于空值 不等于空值	条件选股取空值或不取空值, 多项输入用 and 连接	isnull([s1])and isnotnull([s2])

附注 9 排序表达式和 Top 表达式使用规则：

字段说明	取值格式	取值示例
排序表达式 :对返回的结果进行排序	格式：orderby=[rd ra] , (rd 为降序 , ra 为升序) 支持变量引用, 引用格式为方括号+变量名 多个排序字段间, 以=> 符号分隔	orderby=rd([s1]*2) => ra([s2])

	支持的操作符：rd,ra,算术运算符，“=>”分隔符	
Top 表达式：对返回结果按指定排序提取前 N 行	格式：top=max(排序字段表达式,行数) 或 top=min(排序字段表达式,行数) 支持变量引用，引用格式为方括号+变量名 支持的操作符 and,or,算术运算符	top = max([s1],100) and min([s2],100)
板块成分日期：选择的板块成分的日期	格式：sectordate= 板块历史成分的日期，若 sectordate 不传则默认取最新的一天。	sectordate=2018-07-18

3.2.11 宏观数据函数：edb

edb(edbids, options)				
获取宏观指标数据				
参数名	简称	定义		描述
edbids	宏观指标 id	字符串	输入	宏观指标 id，支持多代码输入，以半角逗号分隔，宏观指标列表见 量化接口官网-命令生成-宏观数据
options	可选参数	字符串	输入	附加参数，可填“ ”，可填附加字段，见 附注 10
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID</pre>			

	<pre> self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果 </pre>
<p>范例 (Python2.x)</p>	<pre> data = c.edb("EMM00087117","IsPublishDate=1") if(data.ErrorCode != 0): print("request edbquery Error, ", data.ErrorMsg) else: print "edbid ", for ind in data.Indicators: print ind, " ", print "" for code in data.Codes: for j in range(0, len(data.Dates)): print code, " ", " ", for i in range(0, len(data.Indicators)): print data.Data[code][i][j], " ", print "" </pre>
<p>范例 (Python3.x)</p>	<pre> data = c.edb("EMM00087117","IsPublishDate=1") if(data.ErrorCode != 0): print("request edb Error, ", data.ErrorMsg) else: print("edbid date ",end="") for ind in data.Indicators: </pre>

	<pre> print(ind, end=" ") print("") for code in data.Codes: for j in range(0, len(data.Dates)): print(code, " ", data.Dates[j], end=" ") for i in range(0, len(data.Indicators)): print(data.Data[code][i][j], end=" ") print("") </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

附注 10 宏观数据函数可选参数列表：

中文名称	英文名称	取值范围	说明
起始日期	StartDate	支持格式：	
截止日期	EndDate	YYYYMMDD , YYYY/MM/DD , YYYY/M/D , YYYY-MM-DD , YYYY-M-D	若StartDate不传,从第一条数据开始返回； 若EndDate不传,返回至最新一条数据； 若都不传则输出全部数据.
最新一条数据	IsLatest	取值范围0,1 缺省值：0	取值0，选定日期范围内数据 取值1，最新一条数据
请求指标真实发布日期	IsPublishDate	取值范围 0,1 缺省值：0	取值 0，不请求 publishdate 取值 1，请求 publishdate 备注：只有部分 EDB 指标有发布日期

3.2.12 宏观指标信息查询函数：edbquery

edbquery(edbids, indicators="", options="")				
获取宏观指标id详情信息				
参数名	简称	定义		描述

edbids	宏观指标 id	字符串	输入	宏观指标 id，支持多代码输入，以半角逗号分隔
indicators	详情字段简称	字符串	输入	详情字段简称，支持多字段输入，以半角逗号分隔，传空或匹配不到时则输出全部字段，详细字段列表见 附注 11
options	可选参数	字符串	输入	附加参数，可填 NULL，可填附加字段
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果</pre>			
范例 (Python2.x)	<pre>data = c.edbquery("EMM00058124,EMM00087117,EMG00147350") if(data.ErrorCode != 0): print("request edbquery Error, ", data.ErrorMsg) else: print "edbid ", for ind in data.Indicators:</pre>			

	<pre> print ind, " ", print "" for code in data.Codes: for j in range(0, len(data.Dates)): print code, " ", " ", for i in range(0, len(data.Indicators)): print data.Data[code][i][j], " ", print "" </pre>
<p>范例</p> <p>(Python3.x)</p>	<pre> data = c.edbquery("EMM00058124,EMM00087117,EMG00147350") if(data.ErrorCode != 0): print("request edbquery Error, ", data.ErrorMsg) else: print("edbid ",end=" ") for ind in data.Indicators: print(ind, end=" ") print("") for code in data.Codes: for j in range(0, len(data.Dates)): print(code, " ", end=" ") for i in range(0, len(data.Indicators)): print(data.Data[code][i][j], end=" ") print("") </pre>

附注 11 宏观指标信息查询函数支持字段列表：

字段简称	中文简称	备注
ID	指标ID	
Name	指标名称	
Unit	单位	
Source	来源	
Region	国家/地区	
Frequency	日期频率	1 日 2 周 3 旬 4 半月 5 月 6 季 7 半年 8 年 9 不定期
Startdate	起始日期	
Enddate	截止日期	
Updatetime	更新时间	

3.2.13 新建组合函数：pcreate

def pcreate(combinCode, combinName, initialFound, remark, options="")				
新建组合				
参数名	参数简称	定义		描述
combinCode	组合代码	字符串	输入	组合代码，英文和数字，最大 10 位
combinName	组合名称	字符串	输入	组合名称
initialFound	初始资金	数字	输入	初始资金，上限 99999999999
remark	组合说明	字符串	输入	组合说明

options	附加参数	字符串	输入	附加参数，可填空字符串，可填附加字符串，见 附注 12
返回值	<p>EmQuantData 为结构体</p> <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果</pre>			
范例	<pre>data = c.pcreate("quant001.PF", "组合牛股", 100000000, "这是一个牛股的组合") if(data.ErrorCode != 0): print("request pcreate Error, ", data.ErrorMsg) else: print("create succeed")</pre>			

附注 12 新建组合函数可选参数列表：

中文名称	英文名称	取值范围	说明
组合类型	combintype	1-4	1、成长型 2、指数型 3、平衡型 4、稳健型
创建公司	createcompany	字符串	默认留空

业绩基准	criteria	1-14 默认 3	1、上证指数 2、深证成指 3、沪深 300 指数 4、上证 A 股指数 5、上证 180 指数 6、上证 50 指数 7、深证 100 指数 8、中小板指 9、中小板综 10、创业板指 11、深证综指 12、三板做市指数 13、基金指数 14、中证 500 指数
------	----------	--------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.2.14 组合账户关系查询函数：pquery

pquery(options="")				
提供组合账户信息相关数据				
参数名	简称	定义		描述
options	附加参数	字符串	输入	附加参数
返回值	EmQuantData 为结构体 <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = dict() #数据结果，详见附注 13 </pre>			

<p>范例 (Python2.x)</p>	<pre> data = c.pquery() if(data.ErrorCode != 0): print("request pquery Error, ", data.ErrorMsg) else: print "[key]:", for index in range(0, len(data.Indicators)): print "\t", data.Indicators[index], print "" for k,v in data.Data.items(): if(key == None): key = k print k,": ", for vv in v: print "\t", vv, print "" </pre>
<p>范例 (Python3.x)</p>	<pre> data = c.pquery() if(data.ErrorCode != 0): print("request pquery Error, ", data.ErrorMsg) else: print("[key]:",end="") for index in range(0, len(data.Indicators)): print("\t", data.Indicators[index],end="") </pre>

	<pre> print("") for k,v in data.Data.items(): if(key == None): key = k print(k," ", end="") for vv in v: print("\t", vv, end="") print("") </pre>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

附注 13 返回数据具体参数明细

参数名	简称	备注
combinstyle	组合方式	1：回测 3：模拟
combinmoneytype	组合基准货币	1：人民币
combinidname	组合名称	
combincode	组合代码	
combincriteria	组合业绩基准	1:上证指数 2:深证成指 3:沪深 300 指数 4:上证 A 股指数 5:上证 180 指数 6:上证 50 指数 7:深证 100 指数 8:中小板指 9:中小板综 10:创业板指 11:深证综指 12:三板做市指数 13:基金指数 14:中证 500 指数

3.2.15 组合批量下单函数：porder

porder(combincode, orderdict, remark="",options="")				
组合批量下单				
参数名	简称	定义		描述
combincode	组合代码	字符串	输入	组合代码
orderdict	下单参数	字典	输入	具体下单信息，见 附注 14
remark	备注信息	字符串	输入	备注信息
options	附加参数	字符串	输入	附加参数，见 附注 15
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>			
范例	<pre>orderdict = {'code':['300059.SZ','600000.SH'], 'volume':[1000,200], 'price':[13.11,12.12], 'date':['2017-08-14','2017-08-24'], 'time':['14:22:18','14:22:52'], 'optype':[1,1], 'cost':[0,3], 'rate':[0,2]}</pre>			

	<pre> #组合下单 data = c.porder(key.PF, orderdict, "this is a test") if(data.ErrorCode != 0): print("porder Error, ", data.ErrorMsg) else: print("order succeed") </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

附注 14 : orderdict 参数定义列表

	参数名	简称	定义	是否必填参数	说明
1	code	代码	char	是	东财代码，格式为'300059.SZ'
2	volume	数量/金额	double	是	股票、场内基金数量（正负表示买入、卖出），场外基金申购定投金额、赎回份额
3	price	价格	single	是	交易价格
4	date	日期	char	是	交易日期，格式为"YYYYMMDD,YYYY/MM/DD,YYYY-MM-DD"
5	time	时间	char	是	交易时间，格式为“hhmmss，hh:mm:ss”（只影响当日交易，历史交易后台默认 150000 录入计算）
6	optype	操作	int	否	1 买入，2 卖出，3 申购，4 赎回 股票交易对应 1、2，场外基金交易对应 3、4
7	cost	费用	single	否	场外基金申购、赎回费用/费率，二选一填写，另一个位置填写 0。 费用为 0，则读取费率
8	rate	费率	single	否	费率为 0，则读取费用 两者都为 0，表示 0 费用、费率 两者都不为 0，默认读取第一个费用

附注 15 : 组合交易附加参数 options 列表

中文名称	英文名称	取值范围	说明
补入现金方式	autoAddCash	0--2，缺省值：0	0：不补充

		1：先扣除可用现金，不足再补充 2：全部外部补充本次批量买入操作所需现金
--	--	-----------------------------------------

3.2.16 组合报表查询函数：preport

def preport(combinCode, indicator, options="")				
查询组合报表信息				
参数名	参数简称	定义		描述
combinCode	组合代码	字符串	输入	组合代码，支持单次查询单个组合的单个报表
indicator	报表名称	字符串	输入	报表名称，hold(持仓查询)，record(交易记录查询)，Contri（业绩贡献-已清仓股票），stagePerf（周期回报-阶段回报），profAna（盈亏分析-区间分析），RiskAna（风险分析），VarAna（VAR 分析），GDaily（组合日报），TDaily（交易日报）
options	附加参数	字符串	输入	附加参数，可填空字符串，可填附加字段，见 指标手册
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>			

	<pre> self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果 </pre>
<p>范例</p> <p>(Python2.x)</p>	<pre> data = c.preport("quant001.PF", "record", "startdate=2017/07/12,enddate=2018/01/15") if(data.ErrorCode != 0): print "request preport Error, ", data.ErrorMsg else: for ind in data.Indicators: print ind, " ", print "" for k in data.Data: for it in data.Data[k]: print it, " ", print "" </pre>
<p>范例</p> <p>(Python3.x)</p>	<pre> data = c.preport("quant001.PF", "record", "startdate=2017/07/12,enddate=2018/01/15") if(data.ErrorCode != 0): print("request preport Error, ", data.ErrorMsg) else: for ind in data.Indicators: </pre>

	<pre> print(ind, end=" ") print("") for k in data.Data: for it in data.Data[k]: print(it, end=" ") print("") </pre>
--	--------------------------------------------------------------------------------------------------------------------------------------

3.2.17 删除组合函数：pdelete

def pdelete(combinCode, options="")				
删除组合				
参数名	参数简称	定义		描述
combinCode	组合代码	字符串	输入	组合代码
options	附加参数	字符串	输入	附加参数，可不填
返回值	<p>EmQuantData 为结构体</p> <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果 </pre>			

范例	<pre> data = c.pdelete("quant001.PF") if(data.ErrorCode != 0): print("request pdelete Error, ", data.ErrorMsg) else: print("delete succeed") </pre>
----	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

3.2.18 取消报价订阅：csqcancel

EQErr csqcancel(EQID serialID);				
取消特定或所有的报价订阅				
参数名	简称	定义		描述
serialID	流水号	数字	输入	传入特定流水号，取消对应的报价订阅； 传入 0，取消所有的报价订阅
返回值	EmQuantData 为结构体 <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 </pre>			
范例	<pre> data = c.csq("002716.SZ", "PRECLOSE,OPEN,HIGH,LOW,NOW,AMOUNT") data = c.csqcancel(data.SerialID) </pre>			

3.2.19 板块函数：sector

sector(pukeycode, tradedate, options=None, *arga, **argb)				
获取Choice金融终端指定系统板块证券代码成分列表， 目前只支持沪深股票的历史成分查询，其他板块只能获取最新成分				
参数	简称	定义		描述
pukeycode	板块代码	const char*	输入	通过 Choice 金融终端-板块管理，查询系统 板块代码
enddate	截止日期	const char*	输入	支持格式： YYYYMMDD，YYYY/MM/DD YYYY/M/D，YYYY-MM-DD，YYYY-M-D
options	可选参数	const char*	输入	附加参数，可填 NULL，可填附加字段
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列</pre>			

	<pre> self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果 </pre>
范例 (Python2.x)	<pre> data = c.sector("2000034138", "2016-04-26") if data.ErrorCode != 0: print "request sector Error, ", data.ErrorMsg else: for code in data.Data: print code </pre>
范例 (Python3.x)	<pre> data = c.sector("2000034138", "2016-04-26") if data.ErrorCode != 0: print("request sector Error, ", data.ErrorMsg) else: for code in data.Data: print(code) </pre>

3.2.20 交易日函数: **tradedates**

tradedates(startdate=None, enddate=None, options=None, *arga, **argb)
获取指定交易市场，指定时间区间的日期序列

参数名	简称	定义		描述
startdate	起始日期	字符串或者 datetime		支持格式： YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
enddate	截止日期	字符串或者 datetime		支持格式： YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
options	可选参数	字符串	In	附加参数，可填 NULL，可填附加字段， 见 附注 16
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体 <pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 </pre>			

	self.Data = list() #数据结果
范例 (Python2.x)	<pre>data = c.tradedates("2016-07-01", "2016-07-12") for item in data.Data: print item</pre>
范例 (Python3.x)	<pre>data = c.tradedates("2016-07-01", "2016-07-12") for item in data.Data: print(item)</pre>

附注 16 交易日函数可选参数列表：

中文名称	英文名称	取值范围	说明
日期周期	Period	1--4，缺省值：1	日期周期：日，周，月，年 分别对应：1，2，3，4
按日期排序	Order	1--2，缺省值：1	升序--1； 降序--2
市场类型	Market	见说明，缺省值： “CNSESH”	CNSESH 上海证券交易所 CNSESZ 深圳证券交易所 HKSE00 香港证券交易所 USSE00 美国证券交易所 USSEND 美国纳斯达克市场 USSENY 纽约证券交易所 CNIBBD 中国银行间债券市场 CNFEBC 渤海商品交易所 CNFEDC 大连商品交易所 CNFESF 上海期货交易所 CNFEZC 郑州商品交易所 CNFF00 中国金融期货交易所 CNSH00 沪股通交易日 CNSHHK 沪港股通交易日 CNSZ00 深股通交易日 CNSZHK 深港股通交易日

3.2.21 偏移 N 天的交易日函数：getdate

getdate(tradedate, offday=0, options=None, *arga, **argb)				
获取指定市场交易日历推算第N个交易日				
参数名	简称	定义		描述
tradedate	交易日期	字符串或者 datetime		支持格式： YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
offday	偏移天数	数字	输入	N=0 时，返回交易日当天； N>0 时，交易日往后取最近第 N 个交易日的日期，若交易日期为最新交易日并 N>0, 则返回最新交易日； N<0,时，交易日往前取最近第 N 个交易日的日期。
options	可选参数	字符串	输入	附加参数，可填附加字段，见 附注 17
arga	可选参数	可变参数		预留
argb	可选参数	可变参数		预留
返回值	EmQuantData 为结构体			

	<pre> class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息 self.Codes = list() #东财代码 self.Indicators = list() #指标简称 self.Dates = list() #日期序列 self.RequestID = 0 #请求 ID self.SerialID = 0 #返回的订阅号 self.Data = list() #数据结果 </pre>
<p>范例</p> <p>(Python2.x)</p>	<pre> data = c.getdate("20160426", -3, "Market=CNSESH") if data.ErrorCode != 0: print "request getdate Error, ", data.ErrorMsg else: for tradedate in data.Data: print tradedate </pre>
<p>范例</p> <p>(Python3.x)</p>	<pre> data = c.getdate("20160426", -3, "Market=CNSESH") if data.ErrorCode != 0: print("request getdate Error, ", data.ErrorMsg) else: print(data.Data) </pre>

附注 17 偏移 N 天函数可选参数列表：

中文名称	英文名称	取值范围	说明
市场类型	Market	见说明，缺省值：“CNSESH”	CNSESH 上海证券交易所 CNSESZ 深圳证券交易所 HKSE00 香港证券交易所 USSE00 美国证券交易所 USSEND 美国纳斯达克市场 USSENY 纽约证券交易所 CNIBBD 中国银行间债券市场 CNFEBC 渤海商品交易所 CNFEDC 大连商品交易所 CNFESF 上海期货交易所 CNFEZC 郑州商品交易所 CNFF00 中国金融期货交易所 CNSH00 沪股通交易日 CNSHHK 沪港股通交易日 CNSZ00 深股通交易日 CNSZHK 深港股通交易日

3.2.22 区间交易日天数函数：tradedatesnum

tradedatesnum(startdate, enddate, options)				
获取指定交易市场，指定时间区间的交易日个数				
参数名	简称	定义		描述
startdate	起始日期	字符串或者 datetime	输入	支持格式： YYYYMMDD，YYYY/MM/DD YYYY/M/D，YYYY-MM-DD，YYYY-M-D
enddate	截止日期	字符串或者 datetime	输入	支持格式：

				YYYYMMDD , YYYY/MM/DD YYYY/M/D , YYYY-MM-DD , YYYY-M-D
options	可选参数	字符串	输入	附加参数, 可填附加字段, 见 附注 17
返回值	EmQuantData 为结构体 <pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>			
范例 (Python2.x)	<pre>data = c.tradedatesnum("2018-01-01", "2018-09-15") if data.ErrorCode != 0: print "request tradedatesnum Error, ", data.ErrorMsg else: print "tradedatesnum=====分割线===== " print data.Data</pre>			
范例 (Python3.x)	<pre>data = c.tradedatesnum("2018-01-01", "2018-09-15") if data.ErrorCode != 0: print("request tradedatesnum Error, ", data.ErrorMsg) else: print("tradedatesnum=====分割线===== ") print(data.Data)</pre>			

3.2.23 设置代理函数：setproxy

setproxy(type, proxyip, port, verify, usr, pwd)				
设置代理				
参数	简称	定义		描述
type	代理类型	数字（整数）	输入	ePT_NONE：不使用代理 ePT_HTTP：HTTP 代理 ePT_HTTPS：HTTPS 代理 ePT SOCK4：SOCK4 代理 ePT SOCK5:SOCK5 代理
proxyip	代理服务 器地址	字符串	输入	代理服务器 IP
port	代理服务 器端口	数字（整数）	输入	代理服务器 Port
verify	是否验证 账户名和 密码	布尔值	输入	True：验证代理服务器账户名和密码 False：不验证
usr	账户名	字符串	输入	代理服务器账户名
pwd	密码	字符串	输入	代理服务器密码
返回值	EmQuantData 为结构体			

	<pre>class EmQuantData: def __init__(self): self.ErrorCode = 0 #错误码 self.ErrorMsg = 'success' #错误信息</pre>
范例	<pre>data = c.setproxy(ePT_HTTP, "120.0.0.0", 8080, True, "choice", "password") if data.ErrorCode != 0: print("setproxy failed, ", data.ErrorMsg)</pre>

3.2.24 配置函数：setserverlistdir

setserverlistdir (serlistpath)				
设置 ServerList.json.e 文件存放目录				
参数名	参数简称	定义		描述
serlistpath	文件路径	字符串		ServerList.json.e 文件存放目录
返回值	无			
范例	setserverlistdir ("../bin");			

4. 错误类型定义列表

错误标识	错误描述	错误代码
EQERR_SECUSS	成功	0

EQERR_NO_LOGIN	用户未登陆	10001001
EQERR_USERNAMEORPASSWORD_ERR	用户名或密码错误	10001002
EQERR_NO_ACCESS	用户无 API 权限	10001003
EQERR_ACCESS_EXPIRE	用户 API 权限过期	10001004
EQERR_GETUSERINFO_FAIL	获取用户信息失败	10001005
EQERR_DLLVERSION_EXPIRE	DLL 版本号过期	10001006
EQERR_NO_LV2_ACCESS	用户无 API_LV2 权限	10001007
EQERR_LV2_ACCESS_EXPIRE	用户 API_LV2 权限过期	10001008
EQERR_LOGIN_COUNT_LIMIT	账号登陆数达到上限	10001009
EQERR_LOGIN_FAIL	用户登陆失败	10001010
EQERR_LOGIN_DISCONNECT	用户登陆掉线	10001011
EQERR_ACCESS_INSUFFICIENCE	用户权限不足	10001012
EQERR_IS_LOGIN	用户正在登录	10001013
EQERR_NEED_ACTIVATE	需要登录激活	10001014
EQERR_LOGIN_SERVICE_ERR	登录服务异常	10001015
EQERR_IS_MANUAL_ACTIVATE	正在人工激活	10001016
EQERR_NOTNEED_MANUAL_ACTIVATE	无需人工激活	10001017

EQERR_MANUAL_ACTIVATE_FAIL	人工激活失败	10001018
EQERR_DIFFRENT_DEVICE	激活设备与登录设备不一致	10001019
EQERR_USERINFO_EXPIRED	userInfo 已失效需重新激活	10001020
EQERR_QUOTE_LOGIN_FAIL	行情服务登录验证失败	10001021
EQERR_QUOTE_FLOW_FAIL	行情服务流量验证失败	10001022
EQERR_GET_TRADE_FAIL	获取交易日失败	10000001
EQERR_INIT_OBTAIN_CLASS_FAIL	初始化主类失败	10000002
EQERR_NEW_MEM_FAIL	申请内存失败	10000003
EQERR_PARSE_DATA_ERR	解析数据错误	10000004
EQERR_UNGZIP_DATA_FAIL	gzip 解压失败	10000005
EQERR_UNKNOWN_ERR	未知错误	10000006
EQERR_FUNCTION_INTERNAL_ERR	函数内部错误	10000007
EQERR_OUTOF_BOUNDS	数组越界	10000008
EQERR_NO_DATA	无数据	10000009
EQERR_SYSTEM_ERROR	系统级别错误	10000010
EQERR_SERVERLIST_ERROR	服务器列表错误	10000011
EQERR_OPERATION_FAILURE	操作失败	10000012

EQERR_SERVICE_ERROR	服务错误	10000013
EQERR_GETSERVERLIST_FAIL	获取服务器列表失败	10000014
EQERR_SERVICE_TIMEOUT	服务超时	10000015
EQERR_FREQUENCY_OVER	请求频次过高	10000016
EQERR_SOCKET_ERR	网络错误	10002001
EQERR_CONNECT_FAIL	网络连接失败	10002002
EQERR_CONNECT_TIMEOUT	网络连接超时	10002003
EQERR_RECVCONNECTION_CLOSED	网络接收时连接断开	10002004
EQERR_SENDSOCK_FAIL	网络发送失败	10002005
EQERR_SENDSOCK_TIMEOUT	网络发送超时	10002006
EQERR_RECVSOCK_FAIL	网络接收错误	10002007
EQERR_RECVSOCK_TIMEOUT	网络接收超时	10002008
EQERR_QUOTE_RECONNECT_FAIL	行情服务器连续重连失败	10002009
EQERR_INPARAM_EMPTY	传入参数为空	10003001
EQERR_OUTPARAM_EMPTY	传出参数为空	10003002
EQERR_PARAM_ERR	参数错误	10003003
EQERR_START_DATE_ERR	起始日期格式不正确	10003004

EQERR_END_DATE_ERR	截止日期格式不正确	10003005
EQERR_START_BIGHTHAN_END	起始日期大于截至日期	10003006
EQERR_DATE_ERR	日期格式不正确	10003007
EQERR_CODE_INVALIDED	无效的证券代码	10003008
EQERR_CODE_REPEAT	证券代码重复	10003009
EQERR_INDICATOR_INVALIDED	无效的指标	10003010
EQERR_USERNAME_EMPTY	用户名为空	10003011
EQERR_PASSWORD_EMPTY	密码为空	10003012
EQERR_TO_UPPER_LIMIT	订阅数或股票总数达到上限	10003013
EQERR_MIXED_INDICATOR	不支持的混合指标	10003014
EQERR_INDICATOR_TO_UPPER_LIMIT	单次订阅指标达到上限	10003015
EQERR_BEYOND_DATE_SUPPORT	超出日期支持范围	10003016
EQERR_BASE_LESS_THAN_END	复权基期小于截止日期	10003017
EQERR_MIXED_CODES_MARKET	不支持的混合证券品种	10003018
EQERR_NO_SUPPORT_CODES_MARKET	不支持的证券代码品种	10003019
EQERR_ORDER_TO_UPPER_LIMIT	交易条数超过上限	10003020
EQERR_NO_SUPPORT_ORDERINFO	不支持的交易信息	10003021

EQERR_INDICATOR_REPEAT	指标重复	10003022
------------------------	------	----------