DataAPI 文档

发布 *0.5.0*

东兴投资

2016 年 01 月 11 日

目录

CHAPTER 1

更新历史

2016 年 1 月 11 日

- [更新]：现在在 index 的行情数据中，同时支持输入指数的原始代码或者 TDB 转换后的代码（使用 TDB 转换代码时，api 会打印警告信息提示）。
- [更新]：股票 EOD 价格数据切换至 WindDB.ASHAREEODPRICES。
- [更新]：指数 EOD 价格数据切换至 WindDB.AINDEXEODPRICES。

2016 年 1 月 7 日

- [新功能]：增加关于主题热门股的 api，来源：通联数据 (https://open.datayes.com) 。

2016 年 1 月 6 日

- [新功能]：增加指数成分股的 api。

2015 年 12 月 31 日

- [新功能]：增加获取股票信息的 api。
- [更新]：股票数据源切换至 patch 表中。

2015 年 12 月 17 日

- [新功能]：增加获取期货合约信息的 api。
- [修正]：在某些情况下，获取复权数据的时候，`closePrice` 会变成非常接近于 0 的错误。

2015 年 12 月 15 日

- [修正]：取消行情数据中填充 nan 数据的操作，需要确定更合乎逻辑的操作流程。

2015 年 12 月 14 日

- [新功能]：在行情 api 中增加参数 `instrumentIDasCol` ，将该参数设为 `True` 可以将 `instrumentID` 转为 column index。

- [修正]：将缓存机制使用的 hdf 文件储存格式从 table 调整为 fixed。

2015 年 12 月 11 日

- [新功能]：增加获取混合品种行情数据的 api。

2015 年 12 月 10 日

- [更新]：将对 `tables` 的依赖改为可选并更新相应文档。

2015 年 11 月 23 日

- [新功能]：完善复权的获取方式，用户可以在 `baseDate` 指定 `start`（后复权）或者 `end`（前复权）。
- [修正]：修改 `chunksize` 不为 0 时的行为，保证输出的数据仍然为按照时间排序的。

2015 年 11 月 20 日

- [修正]：有的时候无法正确识别缓存数据的 bug。

2015 年 11 月 19 日

- [新功能]：现在的行情 API 可以启用本地缓存提高读取速度。默认情况下，缓存机制不开启。用户可以在参数中使用 `forceUpdate=False` 来开启缓存机制。
- [修正]：取消在 `windows` 环境下 `multiprocessing` 模块的使用。该模块会在某些特定情况下引发 `RuntimeError`；
- [修正]：`python3` 下整除问题的修复。

2015 年 11 月 17 日

- [修正]：修改 `_GetBarData` 实现，保证使用 `chunksize` 参数时的行为与不使用时候的一致性。

2015 年 11 月 16 日

- [新功能]：股票行情数据接口支持获取复权数据；

CHAPTER 2

介绍

2.1 目的

提供基于 Python 的数据库访问，主要目的：

1. 将用户从繁琐的拼写 sql 语句的工作中解放出来，更专注于数据具体的内容；
2. 增加抽象层，避免用户对数据库的直接访问。方便未来 DBA 对于数据库的更新、维护以及解耦。

2.2 为 DataAPI 添砖加瓦

数据 API 是一个需求驱动型的项目，单个的 api 往往即对应一个典型的数据库查询情景，例如这条查询语句:

```
select instrumentID, tradingDate, closePrice, openPrice from EQY_EOD
    where instrumentID='600000' and tradingDate > '2015-01-01' and tradingDate < '2015-10-01'
```

等价于如下的 api 函数调用：

```
api.GetEquityBarEOD('600000', '2015-01-01', '2015-10-01', ['closePrice', 'openPrice'])
```

所以，最简单的为 DataAPI 做出贡献的办法是：

小技巧： 把你最最最常用 SQL 语句发给我们!!（不要忘记附上简单的查询含义，这样方便我们进行 api 命名）

CHAPTER 3

安装

3.1 依赖

推荐使用 Anaconda (https://www.continuum.io/downloads) 发行版，降低使用 pip 安装 numpy 以及 pandas 可能遇到的问题：

- decorator
- enum34 (仅 python2)
- numpy
- pandas
- pymssql
- sqlalchemy

可选依赖（用于缓存机制）：

- tables

3.2 下载

DataAPI 的源代码可以在以下 svn 地址获取:

```
svn co https://10.63.6.72/svn/IT/midOffice/MarketData/DataCenter/DataAPI/trunk
```

3.3 安装

在签出 svn 中最新代码之后，运行:

```
cd DataAPI

python setup.py install
```

3.4 运行

运行 python:

```
In [1]: import DataAPI


In [2]: DataAPI.__version__

Out[2]: '0.5.0'
```

CHAPTER 4

api

4.1 基本使用

该模块负责导入希望用户可以直接访问的数据获取函数。一般来说，用户只需要在使用的时候专注于这个模块的函数，而无须关注其他的实现细节。本质上说，该模块的作用是将分散于各处的数据获取函数以统一的方式展示给用户使用。

典型的一个使用 DataAPI 情景如下:

```
In [1]: from DataAPI import api

In [2]: prices = api.GetEquityBarMin5('000001', '2015-10-09', '2015-10-20', ['closePrice'])

In [3]: prices.tail()
Out[3]:
                     closePrice instrumentID tradingDate       tradingTime
timeStamp
2015-10-20 14:35:00       11.26       000001  2015-10-20  14:35:00.0000000
2015-10-20 14:40:00       11.27       000001  2015-10-20  14:40:00.0000000
2015-10-20 14:45:00       11.29       000001  2015-10-20  14:45:00.0000000
2015-10-20 14:50:00       11.28       000001  2015-10-20  14:50:00.0000000
2015-10-20 14:55:00       11.32       000001  2015-10-20  14:55:00.0000000
```

关于每个证券类型具体的 API，请参看：股票、指数、期货 等。

4.2 函数列表

Created on 2015-11-12

@author: cheng.li, weijun.shen

class api.BAR_TYPE

An enumeration.

class api.ASSET_TYPE

An enumeration.

api.GetEquityBarMin1(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权 'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetEquityBarMin5(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'

- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权 'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.**GetEquityBarEOD**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票日线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权。'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetIndexBarMin1(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetIndexBarMin5(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetIndexBarEOD(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数日线线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：‘000300’ 或者 [‘000300’]
- **startDate** – 开始日期，格式：’YYYY-MM-DD’
- **endDate** – 结束日期，格式：’YYYY-MM-DD’
- **field** – 需要获取的字段类型，例如：[’closePrice’]，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.**GetIndexConstitutionInfo**(*instrumentIDList*, *refDate=None*, *field='*'*, *forceUpdate=True*)

获取指定日的指数成分股信息

参数

- **instrumentIDList** – 证券名或者列表，例如：‘000300’ 或者 [‘000300’]
- **refDate** – 基准日，格式：’YYYY-MM-DD’
- **field** – 需要获取的字段类型，例如：[’inDate’]，不填的话，默认获取所有字段；可用的 field 包括：[instrumentID, windCode, conInstrumentID, conWindCode, inDate, outDate]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

api.**GetFutureBarMin1**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：’ZN1502’ 或者 [’ZN1502’, ‘A1502’]
- **startDate** – 开始日期，格式：’YYYY-MM-DD’
- **endDate** – 结束日期，格式：’YYYY-MM-DD’
- **field** – 需要获取的字段类型，例如：[’closePrice’]，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice,

highPrice, lowPrice, closePrice, volume, turnover, matchItems]

- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetFutureBarMin5(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.GetFutureBarEOD(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货日线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据

- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.**GetGeneralBarData**(*instrumentIDList*, *startDate*, *endDate*, *bType*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取一般证券的行情数据

参数

- **instrumentIDList** – 证券列表，例如:['600000.xshg', 'if1512', '000300.zicn']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD
- **bType** – bar 的类型，例如：BAR_TYPE.MIN1
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取股票复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权。'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

api.**GetFutureInstrumentInfo**(*instrumentDescList=None*, *field='*'*, *refdate=None*, *forceUpdate=True*)

获取期货基本信息数据

参数

- **instrumentDescList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']；不分大小写，例如：'a1502' 同'A1502'；支持模糊查询，例如：'if*' 返回所有以'IF' 开头的证券，'*15*'返回'IF1501','ZN1502' 等证券名称中包含 '15' 的证券及其检索信息；不填的话，默认获取所有证券；

- **field** – 需要获取的字段类型，例如：['market', 'cnName']，不填的话，默认获取所有字段；可用的 field 包括：[instrumetID, windCode, market, enName, cnName]
- **refdate** – 指定日期，将查询范围限制于当日依然处于交易状态的期货合约，例如：instrumentDescList = 'IF*', refdate= '2015-11-11' 时，返回 IF1511，IF1512，IF1603，IF1606 四份期货的信息，非交易日时返回空值；不填的话，默认不限制具体某交易日；
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

api.**GetEquityInstrumentInfo**(*instrumentIDList=None*, *boardName=None*, *field='*'*, *refDate=None*, *forceUpdate=True*)

获取股票基本信息数据

参数

- **instrumentIDList** – 证券名称或者列表，例如：'600000' 或者 ['600000', '000001']，默认为 None，查询所有证券
- **boardName** – 上市板块名称或者列表，例如：' 主板' 或者 [' 主板', '创业板']，默认为 None，查询所有板块
- **field** – 需要获取的字段类型，例如：['instrumentID', 'windCode']，不填的话，默认获取所有字段；
- **refDate** – 指定日期，将查询范围限制于当日依然处于上市状态的证券，格式为：YYYY-MM-DD
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

api.**GetThemeInfo**(*themeName=None*, *themeID=None*, *field='*'*, *forceUpdate=True*)

获取相关的主题列表

参数

- **themeName** – 关注的主题名，支持模糊查找，默认查找全部主题
- **themeID** – 关注的主题 id，为整数，与 themeName 二选一输入
- **field** – 需要获取的字段类型，例如：['newsNumPercent']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, isActive, insertTime, updateTime]
- **forceUpdate** –

返回

api.GetThemeHotness(*themeName*, *startDate*, *endDate*, *field='*'*, *forceUpdate=True*)

获取主题热度时间序列

参数

- **themeName** – 主题名称，例如：u' 金融'
- **startDate** – 起始日，格式：YYYY-MM-DD
- **endDate** – 结束日，格式：YYYY-MM-DD
- **field** – 需要获取的字段类型，例如：['newsNumPercent']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, date, newsNum, newsNumPercent]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

api.GetStocksByTheme(*themeName*, *refDate=None*, *field='*'*, *forceUpdate=True*)

获取指定主题相关的股票

参数

- **themeName** – 主题名称，例如：u' 金融'
- **refDate** – 参考日，格式：YYYY-MM-DD
- **field** – 需要获取的字段类型，例如：['instrumentID']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, instrumentID, cnName, exchangeName, score]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

api.GetActiveThemesRelatedStocks(*refDate=None*, *windows=None*, *topThemes=20*, *topStocks=20*, *field='*'*, *forceUpdate=True*)

获取指定数量的热门主题高相关度的股票

参数

- **refDate** – 参考日，格式：YYYY-MM-DD。默认为 None，取当前日期
- **windows** – 参考周期，可填写正数表示日；或者字符串形式的时间长度，例如：1m。默认为 None，只取 1 日
- **topThemes** – 选取排名靠前多少的主题，默认值为 20
- **topStocks** – 在同主题下选取排名靠前多少的股票，默认值为 20

- **field** – 需要获取的字段类型，例如：['instrumentID']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, instrumentID, cnName, date, themeHotness，score]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

CHAPTER 5

证券基本信息

5.1 基本检索

目前，此模块包含以下功能：

- GetFutureInstrumentInfo：获取期货基本信息；
- GetEquityInstrumentInfo: 获取股票基本信息

5.2 获取期货基本信息

5.2.1 基本使用

具体示例如下：

```
In [1]: api.GetFutureInstrumentInfo('IF1606',['market','windCode'])
Out[1]:
  market   windCode instrumentID
0     CF  IF1606.CF       IF1606
```

上面的代码，获取了沪深 1606（代码 IF1606）的交易市场和万得证券代码。

默认情况下，若不输入任何 field 和 refdate 参数，函数将返回所查询期货证券的所有基本信息，并且不受限于查询时刻的交易状态：

```
In [2]: api.GetFutureInstrumentInfo('IF1606')
Out[2]:
    windCode instrumentID market  cnName enName
0  IF1606.CF       IF1606     CF  IF1606
```

若需要限定查询范围于某一具体日期仍处于可交易状态的期货合约，需要在 api 调用中指定 refdate 参数。完整的函数使用说明可以通过 python 的 help 函数查看：

```
help(api.GetFutureInstrumentInfo)
```

```
Help on function GetFutureInstrumentInfo in module DataAPI.InstrumentInfo:

GetFutureInstrumentInfo(instrumentDescList=None, field='*', refdate=None, forceUpdate=True)

  获取期货基本信息数据

  :param instrumentDescList: 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']；不分大小写，
          例如：'a1502' 同'A1502'；支持模糊查询，例如：'if*' 返回所有以'IF' 开头的证券，'*15*' 返回
          'IF1501','ZN1502' 等证券名称中包含'15' 的证券及其检索信息；不填的话，默认获取所有证券
  :param field: 需要获取的字段类型，例如：['market', 'cnName']，不填的话，默认获取所有字段
  :param refdate: 指定日期，将查询范围限制于当日依然处于交易状态的期货合约，例如：instrumentDescList='IF*',
          refdate='2015-11-11' 时，返回 IF1511，IF1512，IF1603，IF1606 四份期货的信息，非交易日时
          返回空值；不填的话，默认不限制具体某交易日
  :param forceUpdate: 当为 True 时强制刷新数据，不使用缓存。默认为 True
  :return: pandas.DataFrame
```

5.2.2 模糊检索期货合约信息

函数支持模糊检索功能，具体使用方法为使用字符 * 替代需要模糊处理的部分。另外，需要查询的证券名称不区分大小写，例如：

```
In [3]: api.GetFutureInstrumentInfo('if*').head()
Out[3]:
    windCode instrumentID market  cnName enName
0  IF1601.CF       IF1601     CF  IF1601
1  IF1602.CF       IF1602     CF  IF1602
2  IF0906.CF       IF0906     CF  IF0906
3  IF0907.CF       IF0907     CF  IF0907
4  IF0908.CF       IF0908     CF  IF0908
```

以上函数用于查询以 **IF** 开头的合约（即所有沪深 300 指数期货合约）。

```
In [4]: api.GetFutureInstrumentInfo('*1512', field = ['windCode','market']).head()
Out[4]:
     windCode market instrumentID
0   AF1512.CF     CF       AF1512
```

```
1  ag1512.SHF    SHF       AG1512
2  al1512.SHF    SHF       AL1512
3  au1512.SHF    SHF       AU1512
4  bb1512.DCE    DCE       BB1512
```

以上函数用于查询以 **1512** 结尾的合约（即所有在 2015 年 12 月到期的合约）。

```
In [5]: api.GetFutureInstrumentInfo('*15*',field = ['windCode','market']).head()
Out[5]:
    windCode market instrumentID
0  a1501.DCE    DCE        A1501
1  a1503.DCE    DCE        A1503
2  a1505.DCE    DCE        A1505
3  a1507.DCE    DCE        A1507
4  a1509.DCE    DCE        A1509
```

以上函数用于查询包含 **15** 字段的合约（即所有在 2015 年到期的合约）。

5.2.3 限定交易日检索合约信息

若以日期格式 YYYY-MM-DD 输入参数 refdate，函数将返回于该日处于可交易状态的相关合约信息（若指定日期为非交易日，没有合约处于可交易状态），例如：

```
In [6]: api.GetFutureInstrumentInfo('if*',refdate = '2015-12-01')
Out[6]:
    windCode instrumentID market  cnName enName
0  IF1512.CF       IF1512     CF  IF1512
1  IF1603.CF       IF1603     CF  IF1603
2  IF1606.CF       IF1606     CF  IF1606
```

以上函数用于查询在 2015 年 12 月 1 日可交易的沪深指数期货合约。

在函数设定中，参数证券名 instrumentDescList 可以缺省，此时将返回满足其它限定条件的全部证券信息：

```
In [7]: api.GetFutureInstrumentInfo(field = 'market',refdate = '2015-12-01').head()
Out[7]:
  market instrumentID
0    DCE        A1601
1    DCE        A1603
2    DCE        A1605
3    DCE        A1607
```

```
4    DCE         A1609
```

以上函数用于查询在 2015 年 12 月 1 日可交易的所有期货合约名称以及其交易场所。

5.3 获取股票基本信息

5.3.1 基本使用

具体示例如下：

```
In [8]: api.GetEquityInstrumentInfo(instrumentIDList='601198',
   ...: field=['cnName', 'listDate', 'listBoardName'])
   ...:
Out[8]:
  cnName  listDate listBoardName
0   东兴证券  20150226            主板
```

5.3.2 按上市板块查找

可以搜索，例如，所有 **创业板**股票：

```
In [9]: api.GetEquityInstrumentInfo(boardName=u'主板',
   ...: field=['instrumentID', 'cnName', 'listBoardName']).tail()
   ...:
Out[9]:
     instrumentID      cnName listBoardName
1991       600335       国机汽车            主板
1992       600010       包钢股份            主板
1993       600527       江南高纤            主板
1994       000699  S*ST 佳纸（退市）            主板
1995       600129       太极集团            主板
```

5.3.3 按日期查找

可以按照日期查询当日处于上市状态的股票。例如查询 2015 年 1 月 1 日处于上市状态的股票：

```
In [10]: api.GetEquityInstrumentInfo(refDate=u'2015-01-01',
   ....: field=['instrumentID', 'cnName', 'listDate']) \
   ....: .sort_values('listDate').tail()
```

```
   ....:
Out[10]:
     instrumentID cnName  listDate
2508       603889   新澳股份  20141231
2563       300412   迦南科技  20141231
2529       603017   中衡设计  20141231
2509       300411   金盾股份  20141231
2546       300410   正业科技  20141231
```

可以尝试换个时间:

```
In [11]: api.GetEquityInstrumentInfo(refDate=u'2015-02-28',
   ....: field=['instrumentID', 'cnName', 'listDate']) \
   ....: .sort_values('listDate').tail()
   ....:
Out[11]:
     instrumentID cnName  listDate
2536       002742   三圣特材  20150217
2565       603118   共进股份  20150225
2520       601198   东兴证券  20150226
2522       603828    柯利达  20150226
2557       603969   银龙股份  20150227
```

5.4 函数列表

Created on Tue Dec 15 09:50:06 2015

@author: cheng.li, weijun.shen, yucheng.lai

DataAPI.InstrumentInfo.**GetEquityInstrumentInfo**(*instrumentIDList=None*, *boardName=None*, *field='*'*, *refDate=None*, *forceUpdate=True*)

获取股票基本信息数据

参数

- **instrumentIDList** – 证券名称或者列表，例如：‘600000’ 或者 [‘600000’, ‘000001’]，默认为 None，查询所有证券
- **boardName** – 上市板块名称或者列表，例如：’ 主板’ 或者 [’ 主板’, ‘创业板’]，默认为 None，查询所有板块

- **field** – 需要获取的字段类型，例如：['instrumentID', 'windCode']，不填的话，默认获取所有字段；
- **refDate** – 指定日期，将查询范围限制于当日依然处于上市状态的证券，格式为：YYYY-MM-DD
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

DataAPI.InstrumentInfo.**GetFutureInstrumentInfo**(*instrumentDescList=None*, *field='*'*, *refdate=None*, *forceUpdate=True*)

获取期货基本信息数据

参数

- **instrumentDescList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']；不分大小写，例如：'a1502' 同'A1502'；支持模糊查询，例如：'if*' 返回所有以'IF' 开头的证券，'*15*'返回'IF1501','ZN1502' 等证券名称中包含 '15' 的证券及其检索信息；不填的话，默认获取所有证券；
- **field** – 需要获取的字段类型，例如：['market', 'cnName']，不填的话，默认获取所有字段；可用的 field 包括：[instrumetID, windCode, market, enName, cnName]
- **refdate** – 指定日期，将查询范围限制于当日依然处于交易状态的期货合约，例如：instrumentDescList = 'IF*', refdate= '2015-11-11' 时，返回 IF1511，IF1512，IF1603，IF1606 四份期货的信息，非交易日时返回空值；不填的话，默认不限制具体某交易日；
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

CHAPTER 6

证券行情

6.1 如何使用

在 DataAPI 中我们设计了如下的函数获取多种证券的行情（证券这里可以是股票、期货或者指数等）：

- GetGeneralBarData

> **警告:** 在使用以上 api 时，所有的证券代码必须带上交易所后缀。例如：‘600000’ 必须写为 ‘600000.xshg’，‘if1512’ 必须写为’if1512.cffex’ 等。

使用方式如下：

```
In [1]: from DataAPI import api

In [2]: data = api.GetGeneralBarData(['600000.xshg', 'if1512.cffex', '000300.zicn'], \
   ...:                              '2015-01-01', \
   ...:                              '2015-10-01', \
   ...:                              api.BAR_TYPE.EOD)
   ...:
```

获取股票的信息：

```
In [3]: data[data['instrumentID'] == '600000'][['instrumentID', 'closePrice']].tail(5)
Out[3]:
           instrumentID  closePrice
timeStamp
2015-09-24       600000       15.27
2015-09-25       600000       15.59
2015-09-28       600000       15.92
2015-09-29       600000       15.97
2015-09-30       600000       16.63
```

获取期货的信息：

```
In [4]: data[data['instrumentID'] == 'IF1512'][['instrumentID', 'closePrice']].tail(5)
Out[4]:
            instrumentID   closePrice
timeStamp
2015-09-24       IF1512  3069.000000
2015-09-25       IF1512  2994.000000
2015-09-28       IF1512  3024.800049
2015-09-29       IF1512  2941.000000
2015-09-30       IF1512  2989.199951
```

获取指数的信息：

```
In [5]: data[data['instrumentID'] == '000300'][['instrumentID', 'closePrice']].tail(5)
Out[5]:
            instrumentID  closePrice
timeStamp
2015-09-24       000300   3284.9991
2015-09-25       000300   3231.9514
2015-09-28       000300   3242.7524
2015-09-29       000300   3178.8544
2015-09-30       000300   3202.9475
```

6.2 函数列表

Created on 2015-12-11

@author: cheng.li, weijun.shen

DataAPI.General.**GetGeneralBarData**(*instrumentIDList*, *startDate*, *endDate*, *bType*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取一般证券的行情数据

参数

- **instrumentIDList** – 证券列表，例如:['600000.xshg', 'if1512', '000300.zicn']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD
- **bType** – bar 的类型，例如：BAR_TYPE.MIN1

- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取股票复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权。'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

CHAPTER 7

股票

7.1 获取基本行情

现阶段，如下 api 用于行情的获取：

- `GetEquityBarEOD`：获取日线数据；
- `GetEquityBarMin1`: 获取 1 分钟线数据；
- `GetEquityBarMin5`: 获取 5 分钟线数据。

以上的数据接口都有类似的使用方式：

```
In [1]: api.GetEquityBarEOD('600000', '2012-10-08', '2015-09-30', \
   ...: ['closePrice', 'openPrice']).head()
   ...:
Out[1]:
            closePrice  openPrice instrumentID tradingDate       tradingTime
timeStamp
2012-10-08        7.32       7.38       600000  2012-10-08  15:00:00.0000000
2012-10-09        7.45       7.35       600000  2012-10-09  15:00:00.0000000
2012-10-10        7.44       7.43       600000  2012-10-10  15:00:00.0000000
2012-10-11        7.41       7.41       600000  2012-10-11  15:00:00.0000000
2012-10-12        7.45       7.43       600000  2012-10-12  15:00:00.0000000
```

上面的代码，获取了浦发银行（代码 600000）从 2012 年 10 月 8 日到 2015 年 9 月 30 日的开盘价和收盘价数据。

默认情况下，行情 api 获取的数据都是未复权的。为了获取复权的数据，需要在 api 调用中指定 `baseDate` 参数，完整的函数使用说明可以通过 `python` 的 `help` 函数查看：

```
help(api.GetEquityBarEOD)
```

```
Help on function GetEquityBarEOD in module DataAPI.Equity:


GetEquityBarMin1(instrumentIDList,
                 startDate,
                 endDate,
                 field="*",
                 chunksize=None,
                 baseDate=None,
                 forceUpdate=True,
                 instrumentIDasCol=False)
    获取股票日线数据

    :param instrumentIDList: 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
    :param startDate: 开始日期，格式：'YYYY-MM-DD'
    :param endDate: 结束日期，格式：'YYYY-MM-DD'
    :param field: 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段
    :param chunksize: 以分段的形式获取 chunksize 大小的数据
    :param baseDate: 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权。
                     'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；
                     'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；
                     'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；
                     None：不复权
    :param forceUpdate: 当为 True 时强制刷新数据，不使用缓存。默认为 True
    :param instrumentIDasCol: 联合使用 field 以及 instrumentIDs 作为 column 的名字
    :return: pandas.DataFrame
```

例如我们以 2015 年 9 月 30 日为基准日，获取之前时间段的数据（即为向前复权方式）：

```
In [2]: api.GetEquityBarEOD('600000', '2012-10-08', '2015-09-30', \
   ...: ['closePrice', 'openPrice'], baseDate='2015-09-30').head()
   ...:
Out[2]:
            closePrice  openPrice instrumentID tradingDate       tradingTime
timeStamp
2012-10-08    6.173933   6.224539       600000  2012-10-08  15:00:00.0000000
2012-10-09    6.283580   6.199236       600000  2012-10-09  15:00:00.0000000
2012-10-10    6.275145   6.266711       600000  2012-10-10  15:00:00.0000000
2012-10-11    6.249842   6.249842       600000  2012-10-11  15:00:00.0000000
```

```
2012-10-12    6.283580    6.266711        600000  2012-10-12  15:00:00.0000000
```

7.2 分批获取数据

有的时候，有获取大量数据的需求，例如拿全市场股票的过去 3 年的分钟线。在这样的情况下，一次获取全部数据并且塞入内存，会对机器产生极大的负担。特别是在一些内存容量比较低的老机器上面，可能会使界面僵死甚至程序崩溃。所以为了缓和这种情况，行情 api 中我们加入了分批次（chunk）获取数据的功能，通过 chunksize 关键字参数实现：

```
In [3]: data = api.GetEquityBarMin5('600000', '2012-10-08', '2015-09-30', \
   ...: ['closePrice', 'openPrice'], chunksize=5)
   ...:

In [4]: data
Out[4]: <generator object _GetBarDataInChunk at 0x000001A85FBD8A98>
```

在这样的情况下，我们看到，现在返回的是一个 python 的生成器（generator ）。用户可以通过遍历的方法，获取全部数据，每次获取 5 个数据点（chunksize=5 ）：

```
In [5]: print(next(data))
                     closePrice  openPrice instrumentID tradingDate  \
timeStamp
2012-10-08 09:30:00        7.39       7.38       600000  2012-10-08
2012-10-08 09:35:00        7.40       7.39       600000  2012-10-08
2012-10-08 09:40:00        7.39       7.41       600000  2012-10-08
2012-10-08 09:45:00        7.39       7.40       600000  2012-10-08
2012-10-08 09:50:00        7.37       7.39       600000  2012-10-08

                          tradingTime
timeStamp
2012-10-08 09:30:00  09:30:00.0000000
2012-10-08 09:35:00  09:35:00.0000000
2012-10-08 09:40:00  09:40:00.0000000
2012-10-08 09:45:00  09:45:00.0000000
2012-10-08 09:50:00  09:50:00.0000000
```

7.3 使用缓存加速获取数据

获取大量的数据，会消耗许多的时间。如果用户需要反复获取相同的数据，总是从远端服务器读取会将大量的时间浪费在传输上，得不偿失。

为了节省这部分无谓的消耗,DataAPI 设计了缓存的机制,这个功能可以通过设置函数参数: forceUpdate 为 False 来开启。在缓存机制开启的状态下，DataAPI 会优先尝试从本地缓存获取数据；如果获取失败，然后才会向远端服务器请求数据。

> **警告:** 缓存机制开启的时候，需要在当前的运行目录下建立缓存文件夹。所以用户需要确保在当前目录下有写权限。如果 DataAPI 发现本地缓存建立失败，会直接尝试从远端获取数据。
> 缓存机制作为一个高级功能，需要本地建立文件夹保存数据，会产生副作用，不适合不了解该功能的用户使用。所以默认情况下缓存机制并不会开启。

注解: 缓存机制对于所有的行情 DataAPI 都是有效的，例如下面的指数 、期货 。

```
# 读取 50 支股票 2015 年的 5 分钟线数据
import time
from DataAPI import api
from AlgoTrading.Data import set_universe
tickers = set_universe('000300.zicn')[:50]

# 第一次读取，尚未建立缓存，数据会从远端服务器获取
start = time.time()
data = api.GetEquityBarMin5(tickers, '2015-01-01', '2015-10-10', forceUpdate=False)
print(u"直接从远程服务器获取耗时: %.2fs" % (time.time() - start))

# 第二次读取，直接使用本地缓存
start = time.time()
data = api.GetEquityBarMin5(tickers, '2015-01-01', '2015-10-10', forceUpdate=False)
print(u"从缓存获取耗时: %.2fs" % (time.time() - start))
```

表 7.1: 缓存性能比较

方式	运行时间（秒）
直接读取	10.18
使用缓存	0.38

如果缓存获取成功的话，运行窗口中会打印类似的语句:

{'args': (['000001.XSHE', '000002.XSHE', '000009.XSHE' ...], '2015-01-01', '2015-10-10', '*', None, None, False), 'name': 'GetEquityBarMin5', 'kwargs': {}} is reading from cach D:\.dstore\bccf5f7f4a3d6fa8b6c4d1ba6476d4e0e533c00b689b19b83a01f843.hdf...

该返回的内容，主要包含两部分的信息：

- 函数签名

  {'args': (['000001.XSHE', '000002.XSHE', '000009.XSHE' ...], '2015-01-01', '2015-10-10', '*', None, None, False), 'name': 'GetEquityBarMin5', 'kwargs': {}}

- 缓存文件位置

  D:\.dstore\bccf5f7f4a3d6fa8b6c4d1ba6476d4e0e533c00b689b19b83a01f843.hdf

具体的缓存文件名与地址与用户的运行环境有关。

7.4 将 instrumentID 转为 column index

上面的 DataAPI 演示中，instrumentID 数据都是出现在 column 中。有些时候，我们希望将 instrumentID 作为 column index 使用，这样方便同一字段的数据在不同证券之间进行比较。在 api 中，通过 instrumentIDasCol 参数来实现该功能。

例如下面的例子：

```
In [6]: data = api.GetEquityBarEOD(['600000.xshg', '000001.xshe'],
   ...:                            '2012-01-01',
   ...:                            '2015-12-12',
   ...:                            field=['openPrice', 'closePrice', 'turnover'],
   ...:                            instrumentIDasCol=True)
   ...:
```

返回的结果：

```
In [7]: data.tail()
Out[7]:
             openPrice        closePrice            turnover
instrumentID    000001 600000     000001 600000       000001      600000
timeStamp
2015-12-07       12.18  18.71      12.15  19.45  488370699.4  1592198369
2015-12-08       12.07  19.28      11.96  20.10  602568625.9  4764109259
2015-12-09       11.92  19.58      11.99  19.23  515596474.9  1776578423
2015-12-10       11.95  19.34      11.96  19.27  502326072.6  1108934588
2015-12-11       11.91  19.01      11.83  18.60  448800941.4  1630390309
```

如果要获取这里面所有证券收盘价:

```
In [8]: data['closePrice'].tail()
Out[8]:
instrumentID  000001  600000
timeStamp
2015-12-07     12.15   19.45
2015-12-08     11.96   20.10
2015-12-09     11.99   19.23
2015-12-10     11.96   19.27
2015-12-11     11.83   18.60
```

如果只是想获取单支证券的所有信息:

```
In [9]: data.loc[:, pd.IndexSlice[:, '600000']].tail()
Out[9]:
             openPrice closePrice    turnover
instrumentID    600000     600000      600000
timeStamp
2015-12-07       18.71      19.45  1592198369
2015-12-08       19.28      20.10  4764109259
2015-12-09       19.58      19.23  1776578423
2015-12-10       19.34      19.27  1108934588
2015-12-11       19.01      18.60  1630390309
```

获取单支证券的开盘价的话:

```
In [10]: data['openPrice', '600000'].tail()
Out[10]:
timeStamp
2015-12-07    18.71
2015-12-08    19.28
2015-12-09    19.58
2015-12-10    19.34
2015-12-11    19.01
Name: (openPrice, 600000), dtype: float64
```

7.5 函数列表

Created on 2015-11-12

@author: cheng.li, weijun.shen

DataAPI.Equity.**GetEquityBarEOD**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票日线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及 None，默认为 None：不复权。'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Equity.**GetEquityBarMin1**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据

- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及None，默认为 None：不复权 'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Equity.**GetEquityBarMin5**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *baseDate=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取股票 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'600000' 或者 ['600000', '000001']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **baseDate** – 获取复权数据时的默认基准日，可填：日期，'start'，'end'，以及None，默认为 None：不复权 'YYYY-MM-DD'：以固定日期为复权基准日。若该日不是交易日，向前调整至最近的交易日；'start'：以 startDate 起始日期为基准日。若该日不是交易日，向前调整至最近的交易日；'end'：以 endDate 结束日期为基准日。若该日不是交易日，向前调整至最近的交易日；None：不复权
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

CHAPTER 8

指数

8.1 行情信息

现阶段，如下 api 用于行情的获取：

- `GetIndexBarEOD`：获取日线数据；
- `GetIndexBarMin1`: 获取 1 分钟线数据；
- `GetIndexBarMin5`: 获取 5 分钟线数据。

8.2 成分股信息

可以通过下面的 api 获取指定指数在某日的成分股信息：

- `GetIndexConstitutionInfo`

  该 api 具有如下的参数要求：

  - `instrumentIDList`: 指数名称或者列表，例如：'000300' 或者 ['000300']
  - `refDate`: 基准日，格式：'YYYY-MM-DD'；不填的话，默认当前日期最新
  - `field`: 需要获取的字段类型，例如：[' inDate']，不填的话，默认获取所有字段

```
In [1]: api.GetIndexConstitutionInfo('000300',
   ...:                              field=['instrumentID', 'conInstrumentID']) \
   ...:                              .tail()
   ...:
Out[1]:
    instrumentID conInstrumentID
295       000300          600837
```

```
296        000300           002292
297        000300           000001
298        000300           600489
299        000300           601919
```

你可以指定具体的日期:

```
In [2]: api.GetIndexConstitutionInfo('000300',
   ...:                               refDate='2015-01-01',
   ...:                               field=['instrumentID', 'conInstrumentID']) \
   ...:                               .tail()
   ...:
Out[2]:
    instrumentID conInstrumentID
295        000300           600166
296        000300           601958
297        000300           600271
298        000300           601333
299        000300           600340
```

8.3 函数列表

Created on 2015-11-12

@author: cheng.li, weijun.shen

DataAPI.Index.**GetIndexBarEOD**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数日线线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段; 可用的 field 包括: [productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Index.**GetIndexBarMin1**(*instrumentIDList*, *startDate*, *endDate*, *field='**', *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Index.**GetIndexBarMin5**(*instrumentIDList*, *startDate*, *endDate*, *field='**', *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取指数 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Index.**GetIndexConstitutionInfo**(*instrumentIDList*, *refDate=None*, *field='*'*, *forceUpdate=True*)

获取指定日的指数成分股信息

参数

- **instrumentIDList** – 证券名或者列表，例如：'000300' 或者 ['000300']
- **refDate** – 基准日，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['inDate']，不填的话，默认获取所有字段；可用的 field 包括：[instrumentID, windCode, conInstrumentID, conWindCode, inDate, outDate]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

CHAPTER 9

期货

9.1 函数列表

Created on 2015-11-12

@author: cheng.li, weijun.shen

DataAPI.Future.**GetFutureBarEOD**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货日线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Future.**GetFutureBarMin1**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货 1 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

DataAPI.Future.**GetFutureBarMin5**(*instrumentIDList*, *startDate*, *endDate*, *field='*'*, *chunksize=None*, *forceUpdate=True*, *instrumentIDasCol=False*)

获取期货 5 分钟线数据

参数

- **instrumentIDList** – 证券名或者列表，例如：'ZN1502' 或者 ['ZN1502', 'A1502']
- **startDate** – 开始日期，格式：'YYYY-MM-DD'
- **endDate** – 结束日期，格式：'YYYY-MM-DD'
- **field** – 需要获取的字段类型，例如：['closePrice']，不填的话，默认获取所有字段；可用的 field 包括：[productID, instrumentID, tradingDate, tradingTime, openPrice, highPrice, lowPrice, closePrice, volume, turnover, matchItems]
- **chunksize** – 以分段的形式获取 chunksize 大小的数据
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True
- **instrumentIDasCol** – 联合使用 field 以及 instrumentIDs 作为 column 的名字

返回 pandas.DataFrame

CHAPTER 10

主题与情感信息

现阶段，如下 api 用于主题信息的获取：

- GetThemeInfo：获取主题列表；
- GetStocksByTheme: 获取主题相关个股；
- GetThemeHotness: 获取主题热度；
- GetActiveThemesRelatedStocks: 获取热门股票热门股。

10.1 获取主题列表

```
In [1]: api.GetThemeInfo(themeName=u'火箭军')
Out[1]:
   themeID themeName           updateTime           insertTime  isActive
0   122275       火箭军  2016-01-11 09:14:23  2016-01-08 09:03:28         1
```

10.2 获取主题相关个股

```
In [2]: api.GetStocksByTheme(themeName=u'火箭军')
Out[2]:
   themeID themeName instrumentID     score exchangeName cnName
0   122275       火箭军        600118  0.161128      上海证券交易所    中国卫星
1   122275       火箭军        600677  0.000000      上海证券交易所    航天通信
2   122275       火箭军        600855  1.000000      上海证券交易所    航天长峰
```

10.3 获取主题热度

```
In [3]: api.GetThemeHotness(themeName=u'长江经济',
   ...:                     startDate='2016-01-01',
   ...:                     endDate='2016-01-08')
   ...:
Out[3]:
    themeID themeName  newsNum        date  newsNumPercent
0       346     长江经济带       45  2016-01-01        0.698432
1       346     长江经济带       13  2016-01-02        0.238008
2       346     长江经济带       34  2016-01-03        0.507690
3       346     长江经济带      169  2016-01-04        0.509666
4       346     长江经济带       97  2016-01-05        0.285748
5       346     长江经济带      157  2016-01-06        0.485167
6       346     长江经济带      232  2016-01-07        0.815437
7       346     长江经济带      341  2016-01-08        1.210293
8      1664      长江经济       45  2016-01-01        0.698432
9      1664      长江经济       13  2016-01-02        0.238008
10     1664      长江经济       34  2016-01-03        0.507690
11     1664      长江经济      169  2016-01-04        0.509666
12     1664      长江经济       97  2016-01-05        0.285748
13     1664      长江经济      157  2016-01-06        0.485167
14     1664      长江经济      232  2016-01-07        0.815437
15     1664      长江经济      341  2016-01-08        1.210293
```

10.4 获取热门股票热门股

```
In [4]: api.GetActiveThemesRelatedStocks(refDate='2015-01-07',
   ...:                                  topThemes=5,
   ...:                                  topStocks=5)
   ...:
Out[4]:
    themeID themeName instrumentID     score        date cnName  themeHotness
0      2210        证券       601601  0.910033  2015-01-07   中国太保          2184
1      2210        证券       601318  0.868038  2015-01-07   中国平安          2184
2      2210        证券       601628  0.859495  2015-01-07   中国人寿          2184
3      2210        证券       600291  0.754283  2015-01-07   西水股份          2184
4      2210        证券       601336  0.385191  2015-01-07   新华保险          2184
```

5	46	银行	601818	1.000000	2015-01-07	光大银行	2017
6	46	银行	000001	0.949553	2015-01-07	平安银行	2017
7	46	银行	601328	0.916780	2015-01-07	交通银行	2017
8	46	银行	601166	0.698657	2015-01-07	兴业银行	2017
9	46	银行	600036	0.669458	2015-01-07	招商银行	2017
10	2443	综合	600212	1.000000	2015-01-07	江泉实业	1352
11	2443	综合	600724	0.997449	2015-01-07	宁波富达	1352
12	2443	综合	600811	0.997192	2015-01-07	东方集团	1352
13	2443	综合	600095	0.995846	2015-01-07	哈高科	1352
14	2443	综合	000507	0.994531	2015-01-07	珠海港	1352
15	1336	汽车	000581	1.000000	2015-01-07	威孚高科	849
16	1336	汽车	600698	0.995900	2015-01-07	湖南天雁	849
17	1336	汽车	000700	0.995507	2015-01-07	模塑科技	849
18	1336	汽车	600805	0.992337	2015-01-07	悦达投资	849
19	1336	汽车	600480	0.992251	2015-01-07	凌云股份	849
20	66	黄金	601899	1.000000	2015-01-07	紫金矿业	843
21	66	黄金	600489	0.923664	2015-01-07	中金黄金	843
22	66	黄金	002237	0.807393	2015-01-07	恒邦股份	843
23	66	黄金	600547	0.779359	2015-01-07	山东黄金	843
24	66	黄金	002155	0.688065	2015-01-07	湖南黄金	843

10.5 函数列表

Created on 2016-1-7

@author: cheng.li, weijun.shen, yuchen.lai

`DataAPI.Themes.GetActiveThemesRelatedStocks`(*refDate=None*, *windows=None*, *topThemes=20*, *topStocks=20*, *field='*'*, *forceUpdate=True*)

获取指定数量的热门主题高相关度的股票

参数

- **refDate** – 参考日，格式：YYYY-MM-DD。默认为 None，取当前日期
- **windows** – 参考周期，可填写正数表示日；或者字符串形式的时间长度，例如：1m。默认为 None，只取 1 日
- **topThemes** – 选取排名靠前多少的主题，默认值为 20
- **topStocks** – 在同主题下选取排名靠前多少的股票，默认值为 20
- **field** – 需要获取的字段类型，例如：['instrumentID']，不填的话，默认获取

所有字段；可用的 field 包括：[themeID, themeName, instrumentID, cnName, date, themeHotness，score]

- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

DataAPI.Themes.**GetStocksByTheme**(*themeName*, *refDate=None*, *field='*'*, *forceUpdate=True*)

获取指定主题相关的股票

参数

- **themeName** – 主题名称，例如：u' 金融'
- **refDate** – 参考日，格式：YYYY-MM-DD
- **field** – 需要获取的字段类型，例如：['instrumentID']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, instrumentID, cnName, exchangeName, score]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

DataAPI.Themes.**GetThemeHotness**(*themeName*, *startDate*, *endDate*, *field='*'*, *forceUpdate=True*)

获取主题热度时间序列

参数

- **themeName** – 主题名称，例如：u' 金融'
- **startDate** – 起始日，格式：YYYY-MM-DD
- **endDate** – 结束日，格式：YYYY-MM-DD
- **field** – 需要获取的字段类型，例如：['newsNumPercent']，不填的话，默认获取所有字段；可用的 field 包括：[themeID, themeName, date, newsNum, newsNumPercent]
- **forceUpdate** – 当为 True 时强制刷新数据，不使用缓存。默认为 True

返回 pandas.DataFrame

DataAPI.Themes.**GetThemeInfo**(*themeName=None*, *themeID=None*, *field='*'*, *forceUpdate=True*)

获取相关的主题列表

参数

- **themeName** – 关注的主题名，支持模糊查找，默认查找全部主题
- **themeID** – 关注的主题 id，为整数，与 themeName 二选一输入

- `field` – 需要获取的字段类型, 例如: ['newsNumPercent'], 不填的话, 默认获取所有字段; 可用的 field 包括: [themeID, themeName, isActive, insertTime, updateTime]
- `forceUpdate` –

返回

CHAPTER 11

面向开发者

11.1 为自定义的数据 api 开启缓存

访问数据库总是一个比较耗时的过程，假设你有如下的一个 python 中的 sql 访问函数：

```
import pandas as pd
import pymssql


def sqlquery():
    conn = pymssql.connect(host=_HOST, user=_USER, password=_PWD, database=_DB, timeout=_TIME_OUT)
    sql = "select * from EQY_EOD"
    return pd.read_sql(sql, conn)
```

这样一次完整的查询，在作者的开发机上大概运行了 50 秒钟（包括数据库访问时间以及返回结果规整为 pandas.DataFrame 的时间）。

一个很自然的想法是将大规模查询的结果在本地保存，下次需要获取同样的数据时，直接在本地获取，而不是访问远程数据库。由于该功能非常常用，所以开发包中提供了一个单独的装饰器 enableCache 来实现这样的功能：

```
import pandas as pd
import pymssql


@enableCache
def sqlquery_cached():
    conn = pymssql.connect(host=_HOST, user=_USER, password=_PWD, database=_DB, timeout=_TIME_OUT)
    sql = "select * from EQY_EOD"
    return pd.read_sql(sql, conn)
```

在经过这样的包装后，我们为原来的函数提供了缓存功能。分别调用该函数两次，第一次会从远程获取

并保存本地，第二次就会直接获取本地缓存。在作者的开发机上运行以下代码：

```
import datetime as dt


start = dt.datetime.now()
print(sqlquery_cached().tail())
print(dt.datetime.now() - start)


start = dt.datetime.now()
print(sqlquery_cached().tail())
print(dt.datetime.now() - start)
```

两次时间比较如下：

表 11.1: 缓存效果

方式	运行时间（秒）
直接读取	52.67
使用缓存	1.98

11.2 中文编码处理

在 python2 中使用 pandas 访问 sql server 数据库的时候，如果带有中文字段的话，可能会产生乱码。像下面的例子：

```
def unicode_error():
    conn = pymssql.connect(host=_HOST, user=_USER, password=_PWD, database=_DB, timeout=_TIME_OUT)
    sql = "select * from ASHAREDESCRIPTION"
    return pd.read_sql(sql, conn)
```

如果直接调用可能返回如下的结果：

```
                                  OBJECT_ID S_INFO_WINDCODE S_INFO_CODE  \
3884  {FFB9AF1E-A93A-4CE9-B0E6-CDF4A387A2E6}       600527.SH      600527
3885  {FFC3045C-29B5-463F-9660-4FF46E3B6065}       000699.SZ      000699
3886  {FFE0A4FC-E8A3-4918-B070-5FED495B76F4}       002011.SZ      002011
3887  {FFE0E7BE-DF71-42CC-92A3-45E1D868E45B}       600129.SH      600129
3888  {FFE43FAE-C990-40CE-88C4-2B268BEC8919}       002233.SZ      002233


         S_INFO_NAME                 S_INFO_COMPNAME  \
3884        ½ÄÏ¸ßÏË        ½ËÕ½ÄÏ¸ßÏË¹É · ÝÓÐÏÞ¹«Ë¾
```

```
3885  S*ST¼ÑÖ½(ÍËÊÐ)       ¼ÑÄ¼Ë¹½ðµØÕiÖ½¹É · ÝÓÐÏÞ¹«Ë¾
3886         ¶Ü°²» · ¾³     Õã½¶Ü°²È¹¤» · ¾³¹É · ÝÓÐÏÞ¹«Ë¾
3887         Ì«¼«¼¯ÍÅ  ÖØÇìÌ«¼«ÊµÒµ(¼¯ÍÅ)¹É · ÝÓÐÏÞ¹«Ë¾
3888         ËþÅÆ¼¯ÍÅ          ¹ã¶«ËþÅÆ¼¯ÍÅ¹É · ÝÓÐÏÞ¹«Ë¾
```

可以注意到，S_INFO_NAME 以及 S_INFO_COMPNAME 列都显示错误。为了正确的显示中文，我们提供了 cleanColsForUnicode 这样的中文处理装饰器：

```
@cleanColsForUnicode
def unicode_corrected():
   conn = pymssql.connect(host=_HOST, user=_USER, password=_PWD, database=_DB, timeout=_TIME_OUT)
   sql = "select * from ASHAREDESCRIPTION"
   return pd.read_sql(sql, conn)
```

现在调用 unicode_corrected 函数，就会显示正常的结果：

```
                                  OBJECT_ID S_INFO_WINDCODE S_INFO_CODE  \
3884  {FFB9AF1E-A93A-4CE9-B0E6-CDF4A387A2E6}       600527.SH      600527
3885  {FFC3045C-29B5-463F-9660-4FF46E3B6065}       000699.SZ      000699
3886  {FFE0A4FC-E8A3-4918-B070-5FED495B76F4}       002011.SZ      002011
3887  {FFE0E7BE-DF71-42CC-92A3-45E1D868E45B}       600129.SH      600129
3888  {FFE43FAE-C990-40CE-88C4-2B268BEC8919}       002233.SZ      002233


     S_INFO_NAME   S_INFO_COMPNAME  \
3884        江南高纤       江苏江南高纤股份有限公司
3885  S*ST 佳纸（退市）      佳木斯金地造纸股份有限公司
3886        盾安环境     浙江盾安人工环境股份有限公司
3887        太极集团  重庆太极实业（集团）股份有限公司
3888        塔牌集团       广东塔牌集团股份有限公司
```

11.3 函数列表

Created on 2015-11-12

@author: cheng.li, weijun.shen

DataAPI.Utilities.**cleanColsForUnicode**(*f*, **args*, ***kwargs*)

装饰器，为返回值为 pandas.DataFrame 的函数进行中文编码的转换

参数

- **f** – 返回值为 pandas.DataFrame 的函数

- **args** – 准备传递给 f 的位置参数
- **kwargs** – 准备传递给 f 的关键字参数

返回 pandas.DataFrame

DataAPI.Utilities.**enableCache**(*f*, **args*, ***kwargs*)

装饰器，为返回值为 pandas.DataFrame 的函数提供本地缓存功能

参数

- **f** – 返回值为 pandas.DataFrame 的函数
- **args** – 准备传递给 f 的位置参数
- **kwargs** – 准备传递给 f 的关键字参数

返回 pandas.DataFrame