

数据库使用简介

V3.1

Monday, March 12, 2018

汤杰强

一、环境配置

Python 环境的配置

1. 安装 Anaconda: 下载 Anaconda 4.0.0 Python 2.7 32 位版本。

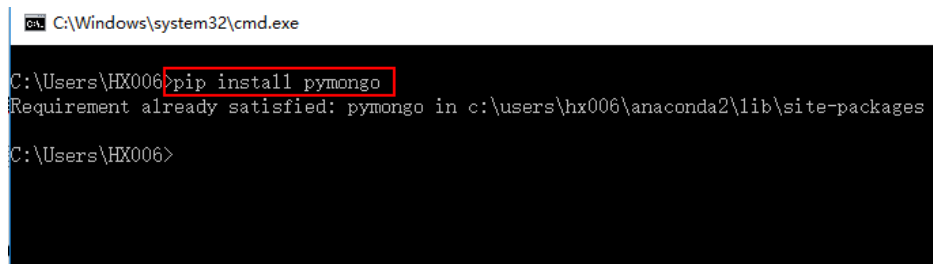
数据库 MongoDB 的配置

1. 安装 MongoDB: 下载 Windows 64-bit 2008 R2+版本
2. 将 MongoDB 注册为 Windows 服务并启动

参考这里

<http://jingyan.baidu.com/article/6b97984dbeef881ca2b0bf3e.html>

3. 安装 mongodb 的 python 接口 pymongo: 在命令行 cmd 中运行 `pip install pymongo`



```
C:\Windows\system32\cmd.exe

C:\Users\HX006>pip install pymongo
Requirement already satisfied: pymongo in c:\users\hx006\anaconda2\lib\site-packages
C:\Users\HX006>
```

4. Robomongo（可选）：MongoDB 的图形化客户端，方便监控和修改数据

The screenshot shows the Robomongo 0.9.0-RC10 application window. The left sidebar displays a tree view of the database structure, including 'New Connection (9)', 'System', 'DAY_MARKET_DB', 'Collections (1)', '000001', 'Functions', 'Users', 'DB', 'FUTURE_MAIN_BAR_DB', 'FUTURE_MAIN_BAR_OI_DB', 'Order_DB', 'TEST_DB', 'VnTrader_1Min_Db', and 'VnTrader_Tick_Db'. The main window shows a query result for the collection '000001' in the 'DAY_MARKET_DB' database. The query is 'db.getCollection('000001').find({})' and the result is a table with 16 rows and 9 columns: _id, date, volume, code, money, high, low, close, and open. The data represents market ticks for a specific code (000001) over time.

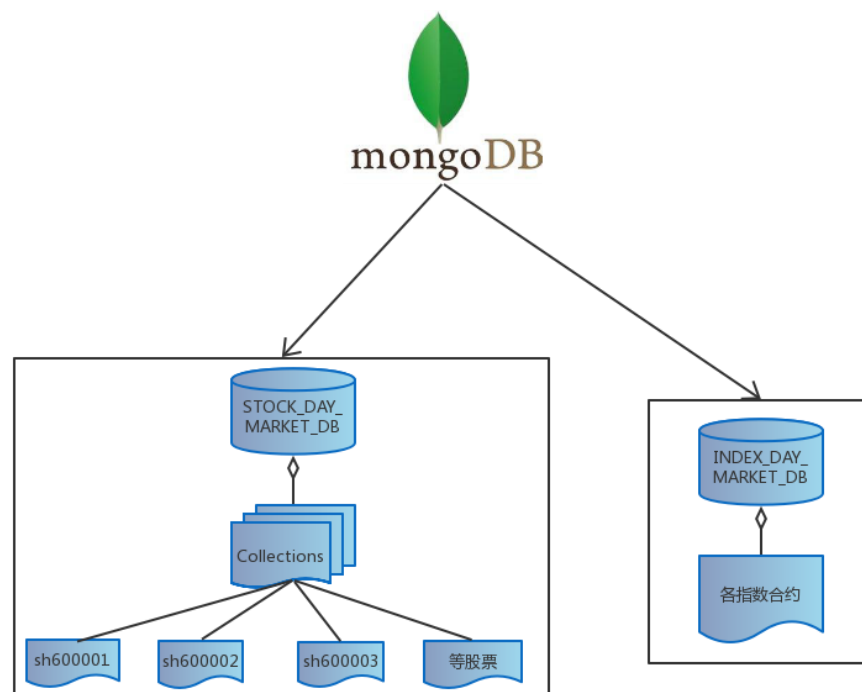
	_id	date	volume	code	money	high	low	close	open
1	ObjectId("...)	2012-01-1...	8780356000	000001	77406846...	2298.376	2196.117	2298.376	2206.531
2	ObjectId("...)	2012-01-1...	8994042200	000001	80655237...	2311.579	2257.898	2266.384	2298.834
3	ObjectId("...)	2012-01-1...	7249886100	000001	63886737...	2305.707	2259.337	2296.075	2266.076
4	ObjectId("...)	2012-01-2...	6992275300	000001	62906415...	2322.891	2293.893	2319.118	2300.5
5	ObjectId("...)	2012-01-3...	5841368700	000001	51868459...	2324.493	2284.291	2285.038	2324.493
6	ObjectId("...)	2012-01-3...	4873832500	000001	43777519...	2296.384	2277.056	2292.61	2285.954
7	ObjectId("...)	2012-02-0...	5355381100	000001	48554972...	2305.863	2263.342	2268.08	2288.066
8	ObjectId("...)	2012-02-0...	6347187200	000001	57682360...	2312.556	2268.687	2312.556	2273.85
9	ObjectId("...)	2012-02-0...	7894369700	000001	70249339...	2336.267	2300.982	2330.405	2306.656
10	ObjectId("...)	2012-02-0...	7573447700	000001	67305626...	2341.575	2317.422	2331.136	2334.248
11	ObjectId("...)	2012-02-0...	6732690600	000001	58091080...	2319.708	2278.825	2291.902	2319.421
12	ObjectId("...)	2012-02-0...	8711452800	000001	80885696...	2350.965	2286.819	2347.53	2291.812
13	ObjectId("...)	2012-02-0...	9803232500	000001	89658695...	2363.613	2333.935	2349.589	2344.224
14	ObjectId("...)	2012-02-1...	9650161900	000001	82222468...	2368.979	2339.885	2351.981	2342.011
15	ObjectId("...)	2012-02-1...	8626358500	000001	77715389...	2365.339	2325.028	2351.855	2332.055
16	ObjectId("...)	2012-02-1...	7313041300	000001	66294933...	2352.469	2332.247	2344.771	2348.877

二、数据库设计

数据说明

主要数据为金融数据，金融数据主要都是时间序列的数据，考虑到这一点，本数据库设计使用 MongoDB 数据库作为本系统所需的数据库，这种非关系型数据库用来存储时间序列数据是非常合适的，而且访问速度比传统的关系型数据库快。该数据库的设计和存储的数据比较单一，所以具有针对性。

数据库结构设计



数据库结构图

在 MongoDB 数据库下分别创建两个数据库，一个是股票日行情数据库 (STOCK_DAY_MARKET_DB)，另一个是各指数日行情的数据库 (INDEX_DAY_MARKET_DB)。

STOCK_DAY_MARKET_DB 数据库

数据库名字：STOCK_DAY_MARKET_DB。

数据库存储内容：各个股票的日行情数据。

各个股票的数据(Collection)名字：各股票的代码。

每条数据包含的信息如下：

STOCK_DAY_MARKET_DB 数据库单条数据属性

属性	说明	备注
code	股票的代码，上证股票以 sh 开头，深证股票以 sz 开头	
date	交易日期	主键
open	开盘价	
high	最高价	
low	最低价	
close	收盘价	
change	涨跌幅，复权之后的真实涨跌幅，保证准确	
volume	成交量	
money	成交额	
traded_market_value	流通市值	
market_value	总市值	
turnover	换手率，成交量/流通股本	
adjust_price	后复权价，复权开始时间为股票上市日，精确到小数点后 10 位	
report_type	最近一期财务报告的类型，3-31 对应一季报，6-30 对应半年报，9-30 对应三季报，12-31 对应年报	此数值可能为空
report_date	最近一期财务报告实际发布的日期	此数值可能为空
PE_TTM	最近 12 个月市盈率，股价 / 最近 12 个月归属母公司的每股收益 TTM	此数值可能为空
PS_TTM	最近 12 个月市销率，股价 / 最近 12 个月每股营业收入	此数值可能为空
PC_TTM	最近 12 个月市现率，股价 / 最近 12 个月每股经营现金流	此数值可能为空
PB	市净率，股价 / 最近期财报每股净资产	此数值可能为空
adjust_price_f	前复权价，复权开始时间为股票最近一个交易日，精确到小数点后 10 位	此数值可能为空

INDEX_DAY_MARKET_DB 数据库

数据库名字：INDEX_DAY_MARKET_DB。

数据库存储内容：各个大盘指数的日行情数据。

各个大盘的数据(Collection)名字：各大盘指数的代码

每条数据包含的信息如下：

INDEX_DAY_MARKET_DB 数据库单条数据属性

属性	说明	备注
code	股票的代码，上证股票以 sh 开头，深证股票以 sz 开头	
date	交易日期	主键
open	开盘价	
high	最高价	
low	最低价	
close	收盘价	
change	涨跌幅，复权之后的真实涨跌幅，保证准确	
volume	成交量	
money	成交额	

三、函数使用

写入数据

股票数据写入

insertStockDayMData(stockCode,filename,dbName=u'STOCK_DAY_MARKET_DB')

函数说明：将 csv 文件中的股票日交易信息，存入到 STOCK_DAY_MARKET_DB 数据库中

stockCode：股票代码如 sh600001，同时这也作为数据库中该股票的 collection

filename：股票相应的 csv 文件的路径，可为绝对路径，也可以为相对路径

dbName：股票数据库名字，默认为 STOCK_DAY_MARKET_DB

例子:

```
code = 'sh600001'
```

```
e = '\\Data\\stock data\\sh600001.csv'
```

```
insertStockDayMData(code,e)
```

_id	date	traded_market_value	volume	market_value	code	money	PC_TTM	PB	PE_TTM	high
1	2009-12-1...	14899055250.0	13100496...	14899055...	sh600001	68934092...	5.190511	1.200098	-673.714678	
2	2009-12-1...	15068042640.0	13070474...	15068042...	sh600001	69096059...	5.249383	1.21371	-681.356086	
3	2009-12-1...	15490511130.0	10496026...	15490511...	sh600001	58460265...	5.396563	1.247739	-700.459724	
4	2009-12-1...	15828485920.0	44208092...	15828485...	sh600001	24844307...	5.514304	1.274962	-715.742253	
5	2009-12-0...	15659498520.0	40122678...	15659498...	sh600001	22493889...	5.455435	1.261351	-708.101133	
6	2009-12-0...	16081967010.0	56619400...	16081967...	sh600001	32353094...	5.602616	1.295381	-727.204804	
7	2009-12-0...	16391777230.0	72415605...	16391777...	sh600001	41827212...	5.710545	1.320335	-741.213743	
8	2009-12-0...	16166460710.0	88570184...	16166460...	sh600001	50217800...	5.632051	1.302186	-731.025442	
9	2009-12-0...	15969308750.0	70685067...	15969308...	sh600001	39627512...	5.563366	1.286306	-722.110268	
10	2009-12-0...	15856650480.0	58221282...	15856650...	sh600001	32732538...	5.524117	1.277231	-717.015871	
11	2009-12-0...	15828485920.0	57674960...	15828485...	sh600001	31972647...	5.514307	1.274963	-715.742565	
12	2009-11-3...	15434182000.0	51363027...	15434182...	sh600001	27720632...	5.376942	1.243203	-697.912982	
13	2009-11-2...	14927219820.0	51752875...	14927219...	sh600001	27904537...	5.200329	1.202368	-674.989005	
14	2009-11-2...	15800321350.0	81553946...	15800321...	sh600001	46674823...	5.504496	1.272694	-714.469141	
15	2009-11-2...	16588929190.0	98972493...	16588929...	sh600001	56456707...	5.779229	1.336215	-750.128762	
16	2009-11-2...	16053803440.0	14813774...	16053803...	sh600001	56350510...	5.503804	1.301113	-735.031304	

数据库中展示

指数数据写入

```
insertIndexDayMData(indexCode,filename,dbName=u'INDEX_DAY_MARKET_DB')
```

函数说明: 将 csv 文件中的大盘指数日交易信息, 存入到 INDEX_DAY_MARKET_DB 数据库中

stockCode: 大盘指数代码如 sh000001, 同时这也作为数据库中指数的 collection

filename: 大盘指数相应的 csv 文件的路径, 可为绝对路径, 也可以为相对路径

dbName: 大盘指数数据库名字, 默认为 INDEX_DAY_MARKET_DB

例子:

```
code = 'sh600001'
```

```
e = '\\Data\\index data\\sh000001.csv'
```

```
insertIndexDayMData(code,e)
```

_id	date	volume	code	money	high	low	close	open	change
4	2017-05-1...	12963523...	sh000001	15016423...	3095.48	3081.28	3090.63	3086.71	0.0001585...
5	2017-05-1...	14862003...	sh000001	16814951...	3103.44	3077.96	3090.14	3082.33	-0.004606...
6	2017-05-1...	16867415...	sh000001	19456059...	3119.58	3101.3	3104.44	3107.8	-0.002736...
7	2017-05-1...	17377556...	sh000001	19329210...	3113.51	3060.53	3112.96	3082.87	0.0073554...
8	2017-05-1...	13566072...	sh000001	15826961...	3098.91	3085.93	3090.23	3085.93	0.0021793...
9	2017-05-1...	15968409...	sh000001	17510725...	3090.49	3051.87	3083.51	3054.11	0.0071892...
10	2017-05-1...	19134190...	sh000001	19760407...	3063.56	3016.53	3061.5	3036.79	0.0028564...
11	2017-05-1...	16079445...	sh000001	18162415...	3090.82	3051.59	3052.78	3078.17	-0.009008...
12	2017-05-0...	13506667...	sh000001	15279590...	3084.21	3056.56	3080.53	3064.85	0.0006236...
13	2017-05-0...	18052690...	sh000001	19817169...	3093.45	3067.69	3078.61	3090.07	-0.007872...
14	2017-05-0...	17621364...	sh000001	20027391...	3117.61	3092.09	3103.04	3114.77	-0.007779...
15	2017-05-0...	17796748...	sh000001	20017085...	3143.82	3111.39	3127.37	3127.11	-0.002545...
16	2017-05-0...	16376392...	sh000001	19023660...	3148.29	3123.75	3135.35	3138.31	-0.002659...
17	2017-05-0...	15422296...	sh000001	17638992...	3154.78	3136.54	3143.71	3147.23	-0.003471...
18	2017-04-2...	16288989...	sh000001	18319577...	3154.73	3136.58	3154.66	3144.02	0.0007835...
19	2017-04-2...	21179307...	sh000001	23574832...	3155.0	3097.33	3152.19	3131.35	0.0036104...

数据库中展示

读取数据

股票数据读取

`loadStockDayM(collectionName,field,startDate,endDate,dbName=u'STOCK_DAY_MARKET_DB')`

函数说明：从数据库中读取相应的数据,返回类型为 list

collectionName: 数据库中的 collection 名字, 及股票的代码如 sh600001

field: 所要取的数据, 如收盘价、成交量等, 详见 *STOCK_DAY_MARKET_DB* 数据库单条数据属性。

startDate: 提取数据的开始时间, 类型为 datetime

endDate: 提取数据的结束时间, 类型为 datetime

dbName: 股票数据库名字, 默认为 STOCK_DAY_MARKET_DB

例子:

`startDate = datetime(2012, 8, 22)`

`endDate = datetime(2013, 9, 25)`

`code = 'sh600001'`

`field = 'close'`

`stockData =loadStockDayM(code,field,startDate,endDate)`

`loadStockDayAllM(collectionName,startDate,endDate,dbName=u'STOCK_DAY_MARKET_DB')`

函数说明：从数据库中读取指定股票时间区间内的所有数据，返回数据类型为 DataFrame

collectionName: 数据库中的 collection 名字，及股票的代码如 sh600001

startDate: 提取数据的开始时间，类型为 datetime

endDate: 提取数据的结束时间，类型为 datetime

dbName: 股票数据库名字，默认为 STOCK_DAY_MARKET_DB

指数数据读取

`loadIndexDayM(collectionName,field,startDate,endDate,dbName=u'INDEX_DAY_MARKET_DB')`

函数说明：从数据库中读取相应的数据,返回类型为 list

collectionName: 数据库中的 collection 名字，及指数的代码如 sh000001

field: 所要取的数据的属性，如收盘价、成交量等，详见 *INDEX_DAY_MARKET_DB 数据库单条数据属性*。

startDate: 提取数据的开始时间，类型为 datetime

endDate: 提取数据的结束时间，类型为 datetime

dbName: 指数数据库名字，默认为 INDEX_DAY_MARKET_DB

例子:

`startDate = datetime(2012, 8, 22)`

`endDate = datetime(2013, 9, 25)`

`code = 'sh000001'`

`field = 'close'`

`indexData =loadIndexDayM(code,field,startDate,endDate)`

`loadIndexDayAllM(collectionName,startDate,endDate,dbName=u'STOCK_DAY_MARKET_DB')`

函数说明：从数据库中读取指定指数时间区间内的所有数据，返回数据类型为 DataFrame

collectionName: 数据库中的 collection 名字，及指数的代码如 sh000001

startDate: 提取数据的开始时间，类型为 datetime

endDate: 提取数据的结束时间，类型为 datetime

dbName: 指数数据库名字，默认为 INDEX_DAY_MARKET_DB

保存数据

股票数据保存

`saveStock2csv(collectionName,savefilename,startDate,endDate)`

函数说明：将所需指定时间区间内的股票数据，保存到相应的 csv 文件中

`collectionName`：数据库中的 `collection` 名字，及股票的代码如 `sh600001`

`savefilename`：数据存放的文件名字，如 `'savedata.csv'`

`startDate`：提取数据的开始时间，类型为 `datetime`

`endDate`：提取数据的结束时间，类型为 `datetime`

例子：

```
startDate = datetime(2012, 8, 22)
```

```
endDate = datetime(2013, 9, 25)
```

```
code = 'sh600001'
```

```
code1 = 'sh600001.csv'
```

```
saveStock2csv(code,code1,startDate,endDate)
```

指数数据保存

`saveIndex2csv(collectionName,savefilename,startDate,endDate)`

函数说明：将所需指定时间区间内的指数数据，保存到相应的 csv 文件中

`collectionName`：数据库中的 `collection` 名字，及指数的代码如 `sh000001`

`savefilename`：数据存放的文件名字，如 `'savedata.csv'`

`startDate`：提取数据的开始时间，类型为 `datetime`

`endDate`：提取数据的结束时间，类型为 `datetime`

例子：

```
startDate = datetime(2012, 8, 22)
```

```
endDate = datetime(2013, 9, 25)
```

```
code = 'sh000001'
```

```
code1 = 'sh000001.csv'
```

```
saveIndex2csv(code,code1,startDate,endDate)
```