Codes Documentation

ANDREW LI

11/26/2019

Description

daily_scraping.py is a multi-processes and multi-threads enabled program for scraping the inforation from Eastmoney forum periodically. This program will be activated twice a day on (1:00 PM and 2:30 PM) and stay silent in background for the rest of the day.

1:00 PM

> The main purpose of this period is to update the database by adding the latest data.
>
> The program will scrape the 300 tickers from CSI300. For each ticker, only one page will be requested at a time. The program will update the historical data in the database by using latest information.
>
> **Time complexity**: more than 25 minutes (depending on the number of total post of that day)
>
> **Concurrency**: 1 Processes and 1 Threads are used, which means that each process is able to send 1 requests to the remote sever at a single time point. After acquiring 1 pages information, the program will take 2 - 3 seconds to parse and reformat the information followed by downloading the rest of tickers.

2:30 PM

> The main purpose of this period is to update the database and create the daily summary table.
>
> The program will scrape the 300 tickers page by page during this period. We will compare the timestamp of scraped page with our historical data to decide whether more pages are needed to be scraped or not.
>
> **Time complexity**: 15 - 20 minutes (depending on the number of post of that day)
>
> **Concurrency**: 1 Processes and 1 Thread are used, which means that each process can only send 1 request to the remote sever at a single time point.

Execution

1. install python3.6 through here
2. run `pip install -r requirements.txt` to install required dependencies
3. run `python -W ignore daily_scraping.py`

Program Explaination

There are 6 main code blocks in `daily_scrapying.py` except for several auxiliary functions.

1. The first block is the *`class stock`* which will be used to create an object for each ticker

```
class Stock:
    """
    Fetch data from http://www.eastmoney.com
    SAMPLE columns:
        -> number of read
    -> comments
    -> title
    -> author
    -> issued time
"""
```