

SCRIPT TO RECORD TICKER DATA from api.pro.coinbase.com

Firstly, this script is meant to be run in Python 3. Once started, it will send a rest request to the api for given currency combination(s) every second. Since the api limits each IP to 3 requests/second, this script handles the scenario and places appropriate time gap between each request. So, if the number of combinations is greater than 3, it sends api request in a burst of 3 combination per second and then waits for a second and resume with rest of the combinations in the same pattern. A separate file for each currency combination, each day is created in the naming pattern 'currency1-currency2-date.csv'.

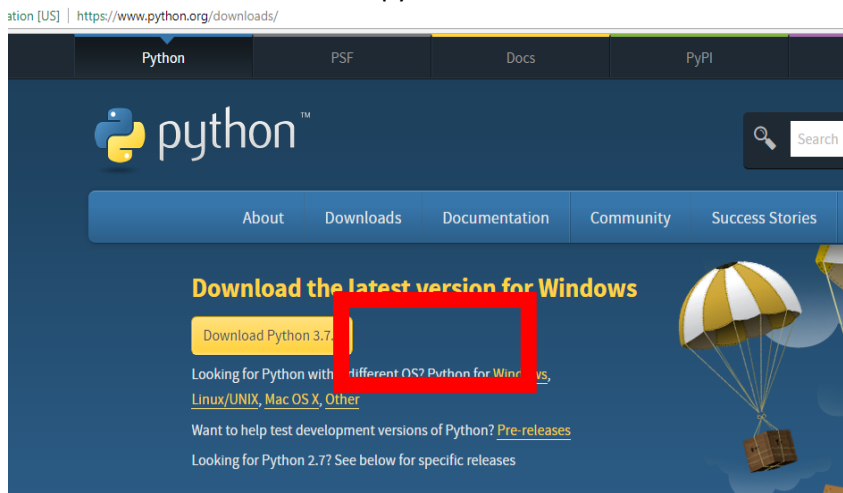
Ex: BTC-EUR-2018-09-06.csv

Additionally, a log file is created every day with logs of outgoing requests(optional) and errors occurred. It is named as 'date_logs.txt'.

Ex: 2018-09-06_logs.txt

DOWNLOADING AND INSTALLING PYTHON:

1. Go to <https://www.python.org/downloads/>
2. Download the latest version of python available:






3. A setup file will be downloaded, which when run will be an easy self-guided installation.
4. Remember the directory in which python is installed, since it needs to be run with the complete path while executing a python script.
5. From here on, python directory would refer to the python installation path: 'E:\Installations\Python' is the path I use and contains all the executables of python.
6. This script uses a python library which sometimes isn't included with the default setup. To ensure that the script runs fine, you need to run an installation command for 'requests' package. This needs to be run from 'cmd' in windows. Go to the python directory in cmd and run the below command:
python -m pip install requests
7. If the library is already installed, it will give the below messages, otherwise, it will add the library in your python installation.

```
E:\Installations\Python>python -m pip install requests
Requirement already satisfied: requests in e:\installations\python\lib\site-packages (2.19.1)
Requirement already satisfied: idna<2.8,>=2.5 in e:\installations\python\lib\site-packages (from requests) (2.7)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in e:\installations\python\lib\site-packages (from requests) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in e:\installations\python\lib\site-packages (from requests) (2018.8.24)
Requirement already satisfied: urllib3<1.24,>=1.21.1 in e:\installations\python\lib\site-packages (from requests) (1.23)
You are using pip version 10.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

CONFIGURATIONS:

Delivered folder contains below files:

Name	Date modified	Type	Size
 get-trade-data-new.py	9/6/2018 6:15 PM	PY File	5 KB
 pairs.csv	9/6/2018 6:00 PM	CSV File	1 KB
 ReadMe.pdf	9/6/2018 7:28 PM	Adobe Acrobat D...	296 KB

'pairs.csv' is the configuration file which needs to be populated before running the script. Please do not rename it or add any additional columns.

	A	B	C	D	E
1	Pairs	timeperiod in days	log interval	requestlog(0 = OFF, 1 = ON)	
2	BTC/USD	1	50	1	
3	BTC/EUR				
4	ETH/BTC				
5	BTC/GBP				
6	ETH/USD				
7	ETH/EUR				
8					
9					
10					
11					
12					
13					
14					
15					

Column A(Pairs):

It needs to be populated with '/' separated currency combinations, with exactly one combination in each cell, starting from A2.

Column B(timeperiod in days – Number only):

Only value of importance here should be in cell B2. This is the time period in days, for which the script will run.

Column C(log interval – Number only):

Again, only one numerical value is expected in this column, in cell C2. This is the burst size of number of request which will be sent after which the data and the logs will be populated. 50 here means, data will be written in the files after every 50 requests. Data/Logs will also be written to files after any failure in request. Also, a separate file will be generated for data and logs after 12:00AM each day.

Column D(requestlog – 0 or 1):

If left on (1), every outgoing api request will be logged. When off (0), only output messages and error messages will be logged.

RUNNING THE SCRIPT:

1. From cmd, go to the script directory (folder where the above provided files reside).
2. Execute the below command to run the script in the foreground:


```
Path_to_python_directory\python.exe get-trade-data-new.py
```

```
E:\Trade-Data-Script\Trade-Data-Script>E:\Installations\Python\python.exe get-trade-data-new.py
Log directory created successfully !
```

For background, just replace *python.exe* with *pythonw.exe*

```
E:\Trade-Data-Script\Trade-Data-Script>E:\Installations\Python\pythonw.exe get-trade-data-new.py
```

3. For foreground, a message saying log 'directory created successfully' will appear on the screen and the script will start processing. If the log directory already exists, it will show the message 'Problem creating log directory!', which just shows that directory already exists and will continue execution.
4. Everything else will now be logged in a file named *date_logs.txt* inside folder *logs* created in the main files' directory.
5. Output files will be created in 'output' folder in the same directory, in the pattern mentioned in the introductory note above.
6. Do not run the script again, if the script is already running. Whether the script is already running or not can be insured by checking the update time for log file or output file. For 50 records as loginterval, file is updated approximately every 3-4 minutes apart. Time of update is proportional to loginterval, hence can be estimated accordingly, if you decide to change it from 50.

Name	Date modified	Type	Size
 2018-09-06_logs.txt	9/6/2018 7:08 PM	Text Document	210 KB

7. You can also check if the script is running, using cmd. From anywhere in cmd, type the command *tasklist*, and a complete list of all the running tasks will appear. You can look for '*python.exe*' or '*pythonw.exe*' in the list, which will signify that the script is running. Please note, that if there are multiple python scripts running in the background, there will be one entry for each running script in the tasklist, and hence it would be impossible to tell them apart using this method.

```
E:\Installations\Python>tasklist
```

chrome.exe	30464	Console	4	47,976 K
EXCEL.EXE	26028	Console	4	109,264 K
SnippingTool.exe	6108	Console	4	19,108 K
python.exe	18288	Console	4	19,012 K
chrome.exe	27824	Console	4	111,860 K
chrome.exe	30224	Console	4	23,116 K
SearchProtocolHost.exe	4896	Console	4	7,820 K

8. To terminate the script, use below command in cmd:

```
Taskkill /PID 18288 /F
```

```
E:\Installations\Python>Taskkill /PID 18288 /F
SUCCESS: The process with PID 18288 has been terminated.
```

The number after PID represents the process ID for your script and can be found from the tasklist above(highlighted).