PROJECT WEEK 3 DATA ANALYST IRONHACK Edwin Pitono

Web Scraping Yahoo Finance

The idea here is to scrap financial data from Yahoo Finance website (I chose page the most active stocks) and try to answer a question

Entrée [3]:
```
# first things first. I found out that the website uses API POST. S
import requests as r
import time
import pandas as pd
```

Entrée [4]:
```
link='https://query1.finance.yahoo.com/v1/finance/screener?crumb=00
```

Entrée [5]:
```
# The pagination is located at the payload. I put time sleep 3 seco

datas=pd.DataFrame()
for i in range(0,9125,25):
    headers="""accept: */*
accept-encoding: gzip, deflate, br
accept-language: en-GB,en-US;q=0.9,en;q=0.8,fr;q=0.7
cache-control: no-cache
content-length: 566
content-type: application/json
cookie: APID=UPd136432b-66b5-11eb-8c62-0291f2e222b6; B=1rmiql5g0u7k
origin: https://finance.yahoo.com
pragma: no-cache
referer: https://finance.yahoo.com/most-active?count=25&offset=25
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: same-site
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 11_2_1) AppleWeb
    payload="""{"offset":"""+str(i)+""","size":25,"sortField":"dayv
    headers= dict(i.split(': ') for i in headers.split('\n'))
    resp=r.post(link, headers=headers, data=payload)
    data=resp.json()['finance']['result'][0]['quotes']
    pd.DataFrame(data)
    print(i," datas are scrapped")
    datas=datas.append(data)
    print("data appended")
    time.sleep(3)
print("Done!")
```

```
data appended
8900  datas are scrapped
data appended
8925  datas are scrapped
data appended
8950  datas are scrapped
data appended
8975  datas are scrapped
data appended
9000  datas are scrapped
data appended
9025  datas are scrapped
data appended
9050  datas are scrapped
data appended
9075  datas are scrapped
data appended
9100  datas are scrapped
data appended
Done!
```

Entrée [72]:
```
datas[datas['symbol']=='MRO'].iloc[:,0:20]
```

Out[72]:

TwoWeekRange	fiftyDayAverageChange	averageDailyVolume3Month	firstTradeDateMilliseconds
3.02 - 9.8	{'raw': 1.2718182, 'fmt': '1.27'}	28389240	138600000.0

Entrée [19]:
```
datas[datas['symbol']=='SABR']
```

Out[19]:

guage	earningsTimestampEnd	regularMarketDayRange	epsForward	regularMarketDayHigh
en-US	{'raw': 1620649800, 'fmt': '2021-05-10', 'long...	{'raw': '13.08 - 14.805', 'fmt': '13.08 - 14.81'}	{'raw': 0.22, 'fmt': '0.22'}	{'raw': 14.805, 'fmt': '14.81'}

Entrée [ ]:
```
#Since we find that many columns has 'raw' and 'fmt', we want to ke
subset = datas['trailingAnnualDividendYield'].copy()
mask = subset.isna()
subset.loc[~mask] = subset.loc[~mask].apply(lambda x: x.get('raw'))
datas['trailingAnnualDividendYield'] = subse
```

Entrée [55]:
```
columns, I select some columns that would be useful for the researcl
'displayName','market','regularMarketPrice','regularMarketDayRange'
```

Entrée [80]:
```
newdata
```

Out[80]:

	Symbol	Stock Name	Country	Market Price	Day Price Range	PE Ratio	marketCap	Dividend Yield	52 W P Ra Cha
0	SNDL	Sundial Growers	us_market	1.530	1.22 - 1.71	NaN	2388865536	NaN	10.086
1	RMM.L	NaN	gb_market	0.340	0.305 - 0.3485	NaN	36326280	0.0222222	0.691
2	VAST.L	NaN	gb_market	0.116	0.1113 - 0.1198	NaN	24708580	NaN	0.183
3	002002.SZ	NaN	cn_market	3.510	3.34 - 3.57	12.4468	9627473920	0.0178042	0.285
4	UJO.L	NaN	gb_market	0.160	0.155 - 0.1687	NaN	31705438	NaN	1.133
...	...	...	...	...	...	...	...	...	
21	CMS.DE	NaN	de_market	18.900	18.9 - 18.9	8.76217	2331598336	0.0253968	0.861
22	6RS.SG	NaN	dr_market	27.400	27.4 - 27.4	20.3415	2119129600	NaN	0.256
23	TZ6.SG	NaN	dr_market	11.200	11.2 - 11.2	NaN	6680766464	NaN	0.866
24	KSD.F	NaN	dr_market	11.666	11.666 - 11.666	NaN	2517662976	NaN	0.143
0	DPR.F	NaN	dr_market	16.500	16.5 - 16.5	71.1207	2796106240	0.0059375	0.571

9101 rows × 13 columns

Entrée [82]:
```
newdata.Country.unique()
```

Out[82]:
```
array(['us_market', 'gb_market', 'cn_market', 'jp_market', 'de_mar
ket',
       'fr_market', 'dr_market'], dtype=object)
```

Entrée [79]:
```
#Then I changed the column names so that they become more comprehen
newdata = newdata.rename(columns={'regularMarketPrice': 'Market Pri
```

Transfering to Mysql

so I created a new database called yahoo_finance in dbeaver

Entrée [57]:
```
import pymysql
from sqlalchemy import create_engine
from getpass import getpass
```

Entrée [59]:
```
username='root'
server='localhost'
database='ironhack'
password=getpass()

engine=create_engine(f'mysql+pymysql://{username}:{password}@{serve
```

```
........
```

Entrée [ ]:
```
#transfering the dataframe and giving it a name yahoo_finance_sql
```

Entrée [83]:
```
newdata.to_sql('yahoo_finance_sql', con=engine,index=False,if_exist
```

Entrée [84]:
```
newdata.to_csv(r'/Users/teahupoo20/Documents/GitHub/data-ft-par-lab
```

Entrée [ ]:
```

```