In [ ]:
```
from bs4 import BeautifulSoup
import requests
import random
import pandas as pd
import xlsxwriter
import time
import sys
```

In [ ]:
```
proxy_list = pd.read_table('proxy2.txt',sep='\t', header=-1)
header_list = pd.read_table('header.txt',sep='\t', header=-1)
```

In [ ]:
```
def reset_proxy():
    rand_sample = random.sample(range(len(proxy_list)), 1)

    rand_ip = proxy_list.ix[rand_sample][0].values
    rand_port = proxy_list.ix[rand_sample][1].values

    proxy_target = (str(*rand_ip) +':'+  str(*rand_port))
#     print(proxy_target)

    return proxy_target
```

In [ ]:
```
def reset_header():
    rand_sample = random.sample(range(len(header_list)), 1)

    rand_header = header_list.ix[rand_sample][0].values

    header_target = (str(*rand_header))
#     print(header_target)

    return header_target
```

In [ ]:
```
url_data = pd.read_table('D:/Crawl/data/call_urls.txt', sep='\t', header=-1)
url_lists = list(url_data[0])
print(url_lists[:5])
print(len(url_lists))
```

In [ ]:
```
base_url = 'https://seekingalpha.com/article/'
```

In [ ]:
```
def save_logs(url_digits):
    log_path = 'D:/Crawl/save_logs.txt'

    with open(log_path, 'a') as f:
        f.write(str(url_digits) + ' : done ' + '\n')
```

In [ ]:
```
def save_error_urls(h, p, url):
    path = 'D:/Crawl/error_logs.txt'

    with open(path, 'a') as f:
        f.write(h + ',' + p + ',' + url + '\n')
```

In [ ]:
```
def get_panels(p_data):
    exa_flag = False
    ana_flag = False

    executives = []
    analysts = []

    for i, p in enumerate(p_data):

        if (p.strong is not None) and (p.text =='Executives'):
            exa_flag = True

        if (p.strong is not None) and (p.text =='Analysts'):
            exa_flag = False
            ana_flag = True

        if (p.strong is not None) and (p.text != 'Analysts') and (p.text != 'Executiv
es'):
            exa_flag = False
            ana_flag = False
            return executives[1:], analysts[1:], i

        if exa_flag:
            executives.append(p.string)

        if ana_flag:
            analysts.append(p.string)
```

In [ ]:
```
def get_body(p_data):
    start_flag = False
    concat_flag = False

    speakers = []
    transcripts = []

    for i,p in enumerate(p_data):

        if (p.strong is not None) and (p.strong.a is None):
            concat_flag = False

        if concat_flag:
            transcripts[-1] = '\n'.join([transcripts[-1], p.text])

        if start_flag:
            transcripts.append(p.text)
            start_flag = False
            concat_flag = True

        if (p.strong is not None) and (p.strong.a is None):
            speakers.append(p.text)
            start_flag = True
            concat_flag = False

    return speakers, transcripts
```

In [ ]:

```
def get_earnings(url):

    headers = {'User-Agent': reset_header()}
    proxy_dict = {'http': reset_proxy()}

    response = requests.get(url, headers=headers, proxies=proxy_dict)
    data_dict = {}

    if (response.status_code == 200):
        soup = BeautifulSoup(response.content, 'html.parser')

        raw_data = soup.find('article')
        header_data = raw_data.find('header')
        data_dict['UploadDate'] = header_data.find('time').get('datetime')
        data_dict['URL'] = header_data.find('meta').get('content')
        data_dict['Title'] = header_data.find('h1').string

        body_data = raw_data.find('div', class_='article-width').findAll('p')
        data_dict['Subtitle'] = body_data[1].text
        data_dict['CallDate'] = body_data[2].text

        executives, analysts, start_point = get_panels(body_data)
        data_dict['Executives'] = executives
        data_dict['Analysts'] = analysts

        parsed_company = body_data[0].text.split('(')

        if len(parsed_company) != 1:
            data_dict['CompanyName'] = parsed_company[0]
            data_dict['Exchange'] = parsed_company[-1].split(':')[0]
            data_dict['Ticker'] = parsed_company[-1].split(':')[1].replace(')', '')
        else:
            data_dict['CompanyName'] = parsed_company[0]
            data_dict['Exchange'] = 'None'
            data_dict['Ticker'] = 'None'

        speakers, transcripts = get_body(body_data[int(start_point):])
        data_dict['Transcript'] = transcripts
        data_dict['Speackers'] = speakers

        return data_dict

    else:
        print(response.status_code, response.body)
        save_error_urls(headers['User-Agent'], proxy_dict['http'], url)
        return None
```

In [ ]:
```
def format_xls(url, data_dict):
    root_path = 'D:/Crawl/'
    path = root_path + 'SeekingAlpha_' + str(url) + '.xlsx'

    workbook = xlsxwriter.Workbook(path)
    worksheet = workbook.add_worksheet('Sheet1') #시트 생성

    start_row = 0
    worksheet.write(start_row,0,'URL')
    worksheet.write(start_row,1, data_dict['URL'])

    start_row += 1
    worksheet.write(start_row,0,'Title')
    worksheet.write(start_row,1, data_dict['Title'])
```

```
    start_row += 1
    worksheet.write(start_row,0,'UploadDate')
    worksheet.write(start_row,1, data_dict['UploadDate'])

    start_row += 1
    worksheet.write(start_row,0,'CompanyName')
    worksheet.write(start_row,1, data_dict['CompanyName'])

    start_row += 1
    worksheet.write(start_row,0,'Exchange')
    worksheet.write(start_row,1, data_dict['Exchange'])

    start_row += 1
    worksheet.write(start_row,0,'Ticker')
    worksheet.write(start_row,1, data_dict['Ticker'])

    start_row += 1
    worksheet.write(start_row,0,'Subtitle')
    worksheet.write(start_row,1, data_dict['Subtitle'])

    start_row += 1
    worksheet.write(start_row,0,'CallDate')
    worksheet.write(start_row,1, data_dict['CallDate'])

    last_row = start_row+1

    num_ex = len(data_dict['Executives'])
    num_an = len(data_dict['Analysts'])
    num_tr = len(data_dict['Transcript'])
#     print(num_ex, num_an, num_tr)

    if (num_ex > 0) :
        for i in range(num_ex):
            worksheet.write(last_row,0, 'Executives')
            worksheet.write(last_row,1, data_dict['Executives'][i])
#             worksheet.write(last_row,1, data_dict['Executives'][i].split(' - ')[0])
#             worksheet.write(last_row,2, data_dict['Executives'][i].split(' - ')[1])
            last_row += 1

    if (num_an > 0) :
        for i in range(num_an):
            worksheet.write(last_row,0, 'Analysts')
            worksheet.write(last_row,1, data_dict['Analysts'][i])
#             worksheet.write(last_row,1, data_dict['Analysts'][i].split(' - ')[0])
#             worksheet.write(last_row,2, data_dict['Analysts'][i].split(' - ')[1])
            last_row += 1

    for i in range(num_tr-1):
        worksheet.write(last_row,0, 'Transcript')
        worksheet.write(last_row,1, data_dict['Speackers'][i])

        if data_dict['Speackers'][i].startswith('Question-'):
            pass
        else:
            worksheet.write(last_row,2, data_dict['Transcript'][i])
        last_row += 1

    workbook.close()
```

In [ ]:
```
print('----------start-------------')

for i, url_digit in enumerate(url_lists[:10]):

    if (i%50 == 0):
        print(i, end=' ')

    url = base_url + str(url_digit)

    try:
        data_dict = get_earnings(url)
        if (data_dict is not None):
            format_xls(url_digit, data_dict)
            save_logs(url_digit)
    except:
        try:
            data_dict = get_earnings(url)
            if (data_dict is not None):
                format_xls(url_digit, data_dict)
                save_logs(url_digit)
        except:
            e = sys.exc_info()[1]
            save_error_urls(str(e), '::', url.split('/')[-1])

    randtime = 0.5 + round(random.random(),2)
    time.sleep(randtime)

print('\n----------end-------------')
```

In [ ]:
```
missed = []

with open('D:/Crawl/last/error_logs.txt', 'r') as f:
    missing = f.readlines()
    for lines in missing:
        if lines.startswith(',,'):
            missed.append(int(lines.replace('\n','').split(',,')[-1]))
        else:
            missed.append(int(lines.replace('\n','').split(',')[-1]))

print(missed[:5])
print(len(missed))

missed = list(set(missed))
print(len(missed))
```