In [ ]:

```
from bs4 import BeautifulSoup
import requests
import random
import pandas as pd
```

In [ ]:

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

base_url = 'https://seekingalpha.com/earnings/earnings-call-transcripts/'
```

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 [ ]:

```
def save_call_urls(filename, sitemap):
    path = 'D:/Crawl/call_urls.txt'
    log_path = 'D:/Crawl/call_logs.txt'

    with open(path, 'a') as f:
        f.write('\n')
        f.write('\n'.join(sitemap))

    with open(log_path, 'a') as f:
        f.write(filename + ' : ' + str(len(sitemap)) + '\n')
```

In [ ]:

```
def save_error_urls(filename):
    path = 'D:/Crawl/error_call_urls.txt'

    with open(path, 'a') as f:
        f.write('\n')
        f.write(filename + ' error ' + '\n')
```

In [ ]:

```
def get_earning_pages(url):

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

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

    if (response.status_code == 200):
#         print(url, ": ", response.status_code)

        soup = BeautifulSoup(response.content, 'html.parser')
        sitemap = []

        for loc in soup.find_all('li', class_='article'):
            if loc.get('data-id') not in sitemap:
                sitemap.append(loc.get('data-id'))
    #         print(len(sitemap))
    #         print(sitemap[:5])

        return sitemap
    else:
        print(response.status_code)
        save_error_urls(url)
        return None
```

In [ ]:

```
import time
import random
```

In [ ]:

```
print('----------start------------')

for i in range(4421):
    print(i, end=' ')
    url = base_url + str(i+1)
    sitemap = get_earning_pages(url)

    if (sitemap is not None):
        save_call_urls('earning-call-transcripts / ' + str(i), sitemap)
    else:
        pass

    randtime = 1 + 1*round(random.random(),2)
    time.sleep(randtime)

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

In [ ]:

```
missed = []

with open('D:/Crawl/data/error_call_urls.txt', 'r') as f:
    missing = f.readlines()
    for lines in missing:
        if lines.startswith('http'):
            missed.append(int(lines.replace('\n','').split('/')[-1].split(' ')[0]))

print(missed)

print('----------start-------------')

for i in missed:
    print(i, end=' ')
    url = base_url + str(i)
    sitemap = get_earning_pages(url)

    if (sitemap is not None):
        save_call_urls('earning-call-transcripts / ' + str(i), sitemap)
    else:
        pass

    randtime = 1 + 1*round(random.random(),2)
    time.sleep(randtime)

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