```python
import
pandas
as pd
    import time
    import bs4
    import random
    import requests
    # !pip install fake-useragent
    from fake_useragent import UserAgent
    import itertools as it

    token = 'https://www.seloger.com/immobilier/locations/immo-paris-75/bien-
    appartement/?LISTING-LISTpg='

    def get_pages(token, nb):
        pages = []
        for i in range(1,nb+1):
            j = token + str(i)
            pages.append(j)
        return pages

    pages = get_pages(token,295)

    # https://www.proxy-list.download/HTTPS
    proxies = pd.read_csv('proxy_list.txt', header = None)
    proxies = proxies.values.tolist()
    proxies = list(it.chain.from_iterable(proxies))

    def get_data(pages,proxies):

        df = pd.DataFrame()
        parameters = ['data-prix','data-codepostal','data-idagence','data-
    idannonce','data-nb_chambres','data-nb_pieces','data-surface','data-typebien']
        ua = UserAgent()
        proxy_pool = it.cycle(proxies)

        while len(pages) > 0:
            for i in pages:
            # on lit les pages une par une et on initialise une table vide pour ranger
    les données d'une page
                df_f = pd.DataFrame()
            # itération dans un liste de proxies
                proxy = next(proxy_pool)
            # essai d'ouverture d'une page
                try:
```

```python
            response = requests.get(i,proxies={"http": proxy, "https": proxy},
headers={'User-Agent': ua.random},timeout=5)
            time.sleep(random.randrange(1,5))
        # lecture du code html et la recherche des balises <em>
            soup = bs4.BeautifulSoup(response.text, 'html.parser')
            em_box = soup.find_all("em", {"class":"agency-website"})
        # extraction des données
            for par in parameters:
                l = []
                for el in em_box:
                    j = el[par]
                    l.append(j)
                l = pd.DataFrame(l, columns = [par])
                df_f = pd.concat([df_f,l], axis = 1)
            df = df.append(df_f, ignore_index=True)
            pages.remove(i)
            print(df.shape)
        except:
            print("Skipping. Connnection error")

    return df


data = get_data(pages,proxies)
```