import requests  
from bs4 import BeautifulSoup  
from time import sleep  
from datetime import datetime  
import pandas as pd  
from tqdm import tqdm  
  
class parsing:  
 # /prodazha/kvartiry/almaty-aujezovskij/?page=  
 # almaty-alatauskij  
 # almaty-almalinskij  
 # shymkent-al-farabijskij  
 def \_\_init\_\_(self, url\_path='/prodazha/kvartiry/almaty-alatauski/?page=', count=3):  
 self.page\_url = 'https://krisha.kz' # указываем локацию  
 result = []  
 for url in tqdm(list(set(self.\_\_get\_urls\_posts(count, url\_path)))):  
 result.append(self.\_\_get\_places\_information(url))  
 sleep(1) #  
 pd.DataFrame(result).to\_csv('dataset\_{0}.csv'.format(datetime.now().strftime("%Y-%m-%d\_%H-%M-%S"))) # эта строка задает имя скачанному файлу с определенным форматом  
 print('Successfully parsed and saved!')  
  
 def \_\_get\_urls\_posts(self, count=3, url\_path='/prodazha/kvartiry/almaty-alatauski/?page='):  
 results = []  
 for i in tqdm(range(2, count)):  
 page = requests.get(self.page\_url + url\_path + str(i) + '/')  
 soup = BeautifulSoup(page.content, 'html.parser')  
 soup\_find = [self.page\_url + page.find(class\_='a-card\_\_title').get('href') for page in  
 soup.find\_all(class\_='a-card\_\_header-left')]  
 sleep(0.7)  
 results += soup\_find  
 return results # в цикле происходит парсинг постов в html коде  
  
 def \_\_get\_places\_information(self, url): # в этом методе идет рапсределение вывода по столбцам по определенным тегам из html кода  
 row = {}  
 page = requests.get(url)  
 soup = BeautifulSoup(page.content, 'html.parser')  
 try:  
 row['title'] = soup.find(class\_='offer\_\_advert-title').find('h1').text  
 except:  
 row['title'] = 'NaN'  
 try:  
 row['price'] = soup.find(class\_='offer\_\_price').text.strip().replace(u'\xa0', '')  
 except:  
 row['price'] = 'NaN'  
 try:  
 for data in soup.find\_all(class\_='offer\_\_info-item'):  
 try:  
 row[data.find(class\_='offer\_\_info-title').text] = data.find(class\_='offer\_\_advert-short-info').text  
 except:  
 pass  
 except:  
 pass  
 try:  
 for data in soup.find(class\_='offer\_\_parameters').find\_all('dl'):  
 try:  
 row[data.find('dt').text] = data.find('dd').text  
 except:  
 pass  
 except:  
 pass  
 try:  
 row['description'] = soup.find(class\_='a-text a-text-white-spaces').text  
 except:  
 row['description'] = 'NaN'  
  
 return row  
  
 def \_\_write\_to\_csv(filename='dataset.csv', row=[]): # внедрение полученных столбцов и строк в скачанный файл  
 with open(filename, 'a') as file:  
 file.write(';'.join(row))  
 file.write('\n')  
#%%time  
p = parsing(count=100)