University of Warsaw
Faculty of Economic Sciences

Kathryn Nagiel
Student's Book Number: 444425

Yufei Sun
Student's Book Number: 426201

Zhao Zhe
Student's Book Number: 433707

Final Project Report for Web Scraping and Social Media Scraping Class

Class taught by
Maciej Wysocki and
Przemyslaw Kurek

Warsaw, May 2022

Project Topic: Otodom Property Listing Scraper

The website we have chosen to scrape is https://www.otodom.pl. On this website, we will filter for "homes for sale in Mazowieckie region" and begin the scraping here.

The information we will be scraping is the title of the listing, the location of the home, and the price using BeautifulSoup, Scrapy, and Selenium. Below is a description of each scraper.

BeautifulSoup

This webscraper is built using requests, time, and pandas packages in python. This scraper works by parsing through the URL below by using a for loop to add the range of 1 through 100 to the end of the URL. "https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/mazowieckie?page="

With this for loop, we will be able to scrape 100 pages of information of houses for sale. We then search for each "box content" of information on the site and then scrape the name of the listing, the location, and price. There is a sleep time of 2 seconds between each page scraped to prevent the website from blocking the IP address if we are loading too many times too quickly. This information is then put into a dictionary and appended to a list called "mazowieckie_for_sale." Then a dataframe is created from this list and exported to an excel file. The excel file contains row numbers, and three columns titled "Title, Location, Price." This excel contains information for 3900 house listings in total, with 39 items being scraped from each page.

To run: type "python3 otodombs.py" in the terminal

Scrapy

This webscraper is built using the python packages requests, scrapy, and BeautifulSoup. For the scrapy webscraper, we used a virtual environment and installed scrapy onto it. Our spider named "allhomes.py" lives in the "homes" folder.

We have our class called "AllhomesSpider" which included the name of the scraper "allhomes", the allowed domains "otodom.pl" and start URL below:
https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/mazowieckie?page=1

We will use a mix of scrapy and BeautifulSoup in this scrapy to gather the information. We first search for the "box" content of each property with the find_all function. Then search each property box and yield the title of listing (name), location, and price. To parse through the next 99 pages, we create a for loop which adds a number to the end of the link below for a range of 2 to 101. For each new link created, we yield the request and of this next page and call the parse function again.

https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/mazowieckie?page=

The spider is called by running: scrapy crawl allhomes -o output.xlsx in the terminal. This runs the spider and then creates an excel file named “output.xlsx” with the information scraped. The excel file contains row numbers, and three columns titled “Title, Location, Price.” This excel contains information for 3900 house listings in total, with 39 items being scraped from each page.

To run: Type “scrapy crawl allhomes -o output.xlsx” in the pycharm terminal.

Selenium

This webscraper is built using the python packages selenium and pandas. This scraper works by parsing through the URL below by using a while loop to add the range of 1 through 100 to the end of the URL.

https://www.otodom.pl/pl/oferty/sprzedaz/mieszkanie/mazowieckie?page=

There is a for loop inside of the while loop to search each information of the property box and yield the title of the name, location, and price. Also, we used the function called ‘find_elements’ to locate the properties from the website that we choose.

The spider is called by running: python3 selenium.py in the terminal. This runs the spider and then creates an excel file named “mazowieckie_for_sale.csv” with the information scraped. The excel file contains row numbers, and three columns titled “Title, Location, Price.” This excel contains information for 3900 house listings in total, with 39 items being scraped from each page.

To run: type “python3 selenium.py” in the terminal