# Example of scrape of two pages with manual URL entry

Our aim is scrape as many For Rent ads as we can to create a list of prospective landlords.

I’ll show a scrape for two examples from Kijiji. Note other websites are encouraged for use. To keep things simple, I’ll take several shortcuts like feed the URLs to be scraped myself instead of automatically and only scraping a few fields.

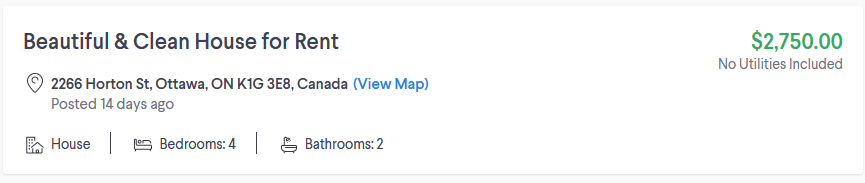
I just want to get a feel for what the end product might look like and the steps to get there.

## Preivew of where we are going

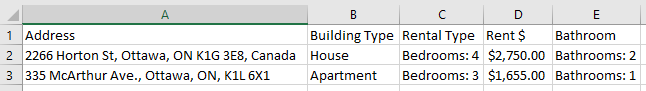
To see the finished product up front, I’ll take the two websites at random from Kijiji below

* <https://www.kijiji.ca/v-apartments-condos/ottawa/beautiful-clean-house-for-rent/1510669287>
* <https://www.kijiji.ca/v-apartments-condos/ottawa/eastwood-park-apartments-3-bedroom-apartment-for-rent/1508660354>

Kijiji happens to organize its information like the screen cap below, but only if the landlord completes the approrpiate form. Some are only presented as raw text, though these may just need to be dealt with on a case by case basis if there aren’t many of them.



The goal is to produce a worksheet that will look something like this (note this leaves some data cleaning steps unfinished and tries to use [the field names our client provided](https://public.3.basecamp.com/p/5KcRbBj2gFpYEuwF11vH9uin) as much as possible):



Using a python script like this:

# First time setup ------------------------------

# pip install beautifulsoup4

# pip install lxml

# pip install future

# Script ----------------------------------------

from bs4 import BeautifulSoup

import csv

# Place a copy of example website in same folder to scrape them locally

page\_file = ["Beautiful & Clean House for Rent Long Term Rentals Ottawa Kijiji.htm", "Eastwood Park Apartments - 3 Bedroom Apartment for Rent Long Term Rentals Ottawa Kijiji.htm"]

# extract fields we want

with open("landlords.csv", "w", newline='') as csvfile:

    f = csv.writer(csvfile)

    # using the field names from the template from Alliance to End Homelessness found in https://public.3.basecamp.com/p/5KcRbBj2gFpYEuwF11vH9uin

    f.writerow(["Address", "Building Type", "Rental Type", "Rent $", "Bathroom"])

    for page in page\_file:

        soup = BeautifulSoup(open(page), features="lxml")

        try:

            # get address

            address = soup.find("span", {"class": "address-3617944557"}).get\_text()

            # get house type, # bedrooms, and # bathrooms

            rooms = soup.findAll("span", {"class": "noLabelValue-3861810455"})

            house\_type = rooms[0].get\_text()

            bedrooms = rooms[1].get\_text()

            bathrooms = rooms[2].get\_text()

            # get price

            price = soup.find("div", {"class": "priceWrapper-1165431705"}).findChild().get\_text()

        except:

            print("bad string{}".format(page))

            continue

        f.writerow([address, house\_type, bedrooms, price, bathrooms])

Note this does not automate generating the URLs in the first place. This type of crawling activity can accomplished with a home brewed script or by using a ready made framework such as https://scrapy.org/.