

VBA Final Project: Home Finder

Bryson Loughmiller
IS 520

12/10/2015

Executive Summary

The Problem

For anyone involved in real estate, whether a mogul looking to buy several properties, or just simply a person trying to find a home to purchase, hours and hours can be wasted looking through the hundreds of listings on the internet. They also have to run the same searches over and over again, typing in the same information that they previously entered in. Then, they have to click on each of the listings to see the details of each home, only to discover that it is not what they are looking for. For almost all sites with housing listings, there is no easy way to get a simple table of data from the search results that allows them to search through to identify a suitable house for purchase.

The Solution

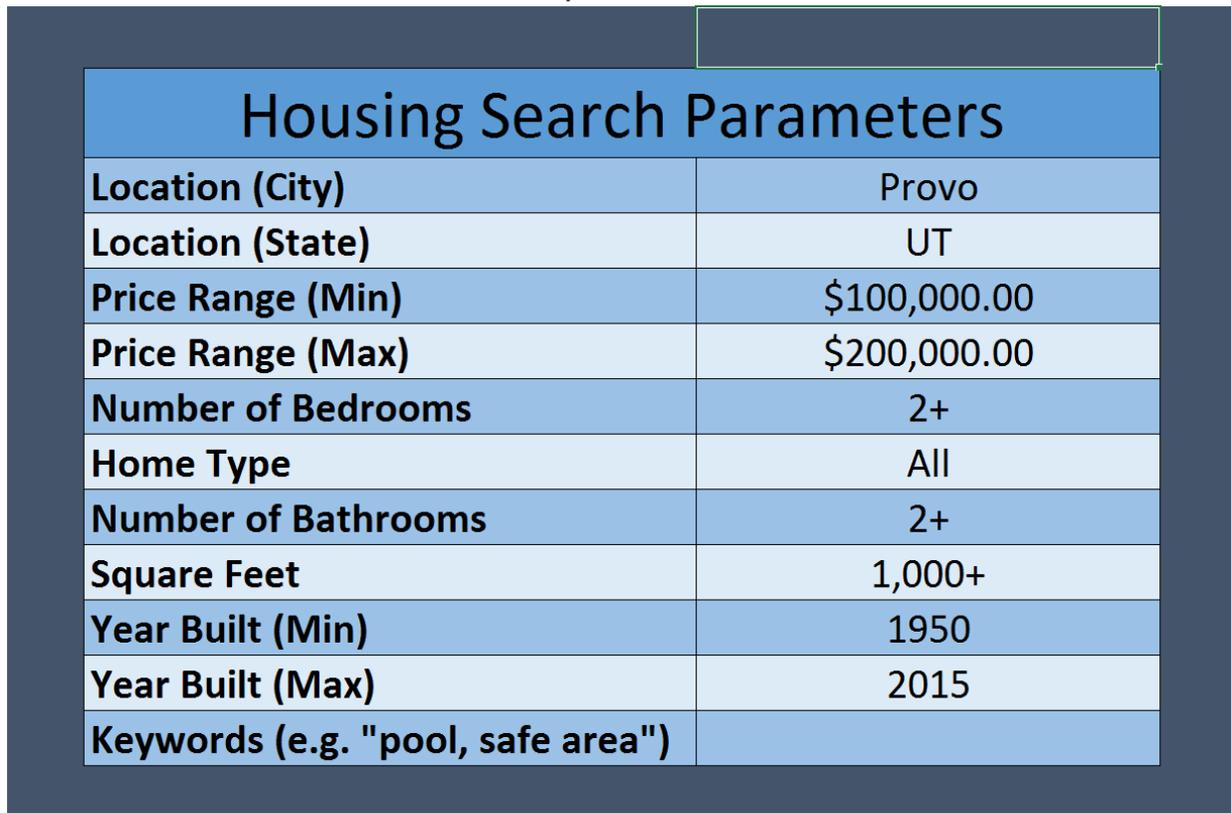
The solution developed here is a VBA-enabled Excel workbook that allows for a user to easily pull a table of all the for sale home listings from KSL within specified parameters. On the "Search Parameters" tab, the user simply enters in the main search parameters for the type of house they are looking for. Once those parameters are entered, the user clicks the "Find New Listings" button in the "Home Finder" tab. Then, in the background, the VBA code goes to KSL.com, runs the search, and goes to the page for each of the listings and scrapes the details for each. This then populates the "Listings" tab with the details, which can then be easily sorted through viewed.

Implementation Documentation

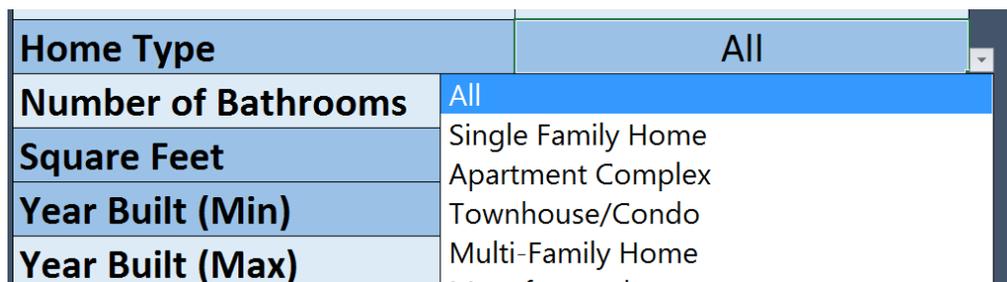
Instructions for use

This workbook is designed to be as easy to use as possible. The user need only follow three steps:

1. Fill out the desired search information in the "Search Parameters" tab of the workbook, as shown in the screenshot below:



Housing Search Parameters	
Location (City)	Provo
Location (State)	UT
Price Range (Min)	\$100,000.00
Price Range (Max)	\$200,000.00
Number of Bedrooms	2+
Home Type	All
Number of Bathrooms	2+
Square Feet	1,000+
Year Built (Min)	1950
Year Built (Max)	2015
Keywords (e.g. "pool, safe area")	

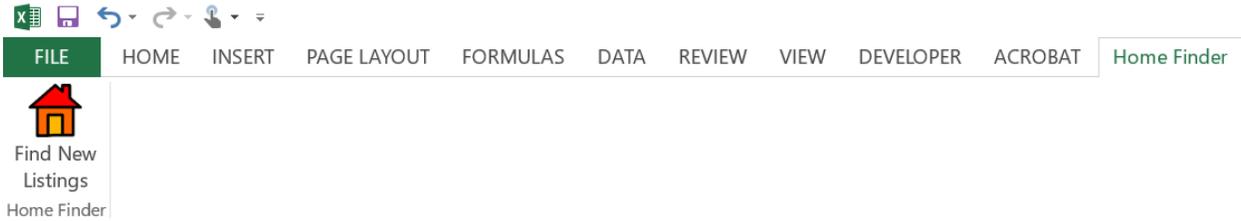


Home Type	All
Number of Bathrooms	All
Square Feet	Single Family Home
Year Built (Min)	Apartment Complex
Year Built (Max)	Townhouse/Condo
	Multi-Family Home

Certain fields are also designed as dropdowns, to ensure valid inputs, as shown below:

(Note: The search parameters data is gathered in the spreadsheet, rather than through a form, so that the user does not have to re-enter data each time they run the search)

- Once the user has finished inputting their search parameters, they click the “Find New Listings” button under the “Home Finder” ribbon:



- The user then simply waits for the search to finish, and can view the listings populating the field as the program continues:

Address	Price	Days on Sit	Property Type	Seller Typ	Bedroom	Bathroom	Year Built	Square Feet	Acres	Heatin	Coolin	Garage/Parkin	Year	Appliances Include	Basement Typ	Floor Cov
841 N 2750 W Provo, UT 84601	\$179,900	2 Hours	Single Family Home	By Agent	4	2	1981	1632	0.29 Acres	Forced Air	Evaporativ	Driveway		Landscap	None Specified	None Spec
437 BELMONT PL #359 Provo, UT	\$176,500	3 Days	Single Family Home	By Agent	3	2	1993	1300	0.03 Acres	Forced Air	Central Air	1 Car	Attached	Gar	Not App	None Spec

A status message in the lower left hand of the workbook also indicates how close the search is to completion with a “% complete” message:

Getting Listings: 78% complete

- The user can then view and sort the data in order to more easily view the available listings.

Current Housing Listings Within Search Parameters

Address	Price	Days on Sit	Property Type	Seller Typ	Bedroom	Bathroom	Year Built	Square Feet	Acres	Heatin	Coolin	Garage/Parkin	
841 N 2750 W Provo, UT 84601	\$179,900	2 Hours	Single Family Home	By Agent	4	2	1981	1632	0.29 Acres	Forced Air	Evaporativ	Driveway	
437 BELMONT PL #359 Provo, UT	\$176,500	3 Days	Single Family Home	By Agent	3	2	1993	1300	0.03 Acres	Forced Air	Central Air	1 Car	Attached
331 Belmont Pl Unit 233 Provo, U	\$176,000	6 Days	Townhouse/Condo	By Agent	3	2	2000	1205	Not Specific	Forced Air	Central Air	1 Car	Carport
605 S 500 W #30 Provo, UT 84601	\$130,000	8 Days	Single Family Home	By Agent	3	2	1995	1004	0.02 Acres	Other	Other	Other	
949 N University Ave Provo, UT 8	\$159,900	9 Days	Townhouse/Condo	By Agent	2	2	1990	850	0.02 Acres	Forced Air	Central Air	2 Car	Attached
1472 E 520 S Provo, UT 84606	\$184,900	9 Days	Single Family Home	By Agent	3	2	1959	1900	0.18 Acres	Forced Air	Central Air	1 Car	Detach
2742 Edgewood Dr Provo, UT 846	\$197,900	30 Days	Townhouse/Condo	By Owner	3	3	1974	1675	0.01 Acres	Forced Air	Central Air	1 Car	Attached
437 Belmont Pl Unit 360 Provo, U	\$177,500	51 Days	Townhouse/Condo	By Agent	3	2	2000	1205	0.03 Acres	Forced Air	Central Air	Street	Parkin
1346 S 1370 E Provo, UT 84606	\$186,900	55 Days	Townhouse/Condo	By Owner	3	2.75	2001	2100	0.13 Acres	Forced Air	Central Air	2 Car	Carport
556 W 800 N Apt 15 Provo, UT 84	\$119,900	58 Days	Townhouse/Condo	By Agent	3	2.5	1979	1700	0.01 Acres	Forced Air	Central Air	Other	
71 S 1530 W Provo, UT 84601	\$199,500	76 Days	Townhouse/Condo	By Owner	3	2	2002	1700	0.06 Acres	Forced Air	Central Air	2 Car	Attached
865 N 500 W Apt 9 Provo, UT 846	\$142,000	106 Days	Townhouse/Condo	By Agent	3	2	1965	991	0.01 Acres	Forced Air	Central Air	Other	

Implementation – Overview of Code Functionality

The following describes the steps that the code takes to gather the data from KSL:

1. Gather input
 - a. The data from the “Search Parameters” tab is gathered and placed into previously declared variables.
2. Generate URL
 - a. The KSL search uses parameters in the URL to determine which listings are shown. Because of this, I was able to create a URL which incorporates the inputs gathered in the above step.
3. Clear Listings table
 - a. The current listings in the “Listings” table are cleared, so that no duplicates are created by the new search.
 - b. The “Listings” tab is selected so that the user can see the listings as they are parsed in.
4. Open Internet Explorer object with search URL
 - a. Using the URL generated above, a background Internet Explorer window is opened which runs the search. The program waits for the search to complete, and then continues.
5. Find all “listing” div tags
 - a. The page is parsed to find all the div tags with class name “listing”.
6. Open listing detail page in new window and gather details
 - a. For each listing found, a new background Internet Explorer window is opened for the detail page of that listing. The details about that listing are gathered and placed in variables.
 - b. The data from those variables is placed in the table on the “Listings” tab
 - c. After the data is placed, the background Internet Explorer process is closed and the next listing is opened.
 - d. This portion takes a significant amount of time, because of the number of internet explorer calls that must be made. Because this takes so long, the status bar on the lower left is updated as the program progresses. This allows the user to see how long is left in the process.
7. Close the original Internet Explorer background process (for the original search)

Lessons Learned and Difficulties Encountered

Concepts Learned

Performing this project allowed me to gain a firmer grasp on a number of key VBA principles that I will be able to use in future projects:

- Web Automation and HTML Parsing
 - Before this class and project, I was unfamiliar with how to use the Microsoft Internet Controls reference to open a background Internet Explorer window and browse the web and gather data. Doing this project allowed me to understand how to better implement this functionality.
 - I was also unfamiliar with HTML parsing. In previous projects that I have done, I would simply import the string and use regular expressions to find the necessary data. However, using the Microsoft HTML Object Library, I was able to easily pull in the desired elements, and search through the children of those elements to find the desired data.
- Creation of Customized Ribbons
 - Before this class, I was totally unaware of the possibility of creating customizable ribbons. We learned the concept in class, but I didn't completely understand how to customize it to my own needs until completing this project. Now I understand how to make the different parts of the ribbon and connect those buttons to the code.

Difficulties Encountered

(NOTE: My original project proposal was different than the project that was completed. I talked with Dr. Allen in person about a new project, one that would scrape housing listings from a website and bring them in, and he approved the new project.)

While performing this project, I encountered a number of difficulties and stumbling blocks.

When I first set out to do this project, I wanted to do it with Zillow.com housing listings, since it is a nationally used website. I spent several hours trying to manipulate links, or use internet explorer to automate the searches, and I got close several times, but I struggled to complete the task. In addition, the html elements each had different, randomly generated class names, which made it incredibly difficult to scrape. I even looked into using the Zillow API to pull the search results, but none of their API parameters would allow for the full search that I was looking for. After spending several hours trying to make the project work with Zillow, I began to look at other sites. KSL formats their elements and search parameters in a nice way that is much easier to scrape and manipulate, so I ended up going with KSL.

I also wanted to add some additional functionality to the project, but was unable to complete the code because I ran out of time. The unfinished code is still included in the project, it is just commented out at the end. The additional feature would have allowed a user to use the workbook on a machine that is always on, like a server of some sort, where it would check for new listings on KSL periodically that matched the search parameters. If a new listing was found, it would send an email to the user about the new listing. The user would be able to enter a time period for checking, such as every two hours. This would have been useful because it would have given the user the opportunity to be one of the first to see new listings, giving them the opportunity to jump on good deals before others. I may have mentioned this portion to Dr. Allen when discussing the new project, but I don't remember if it was actually a part of the project scope or not.

Moving Forward

In the future, I would like to finish implementing the above-mentioned functionality. I would also like to expand this search to allow a user to also look through rental listings, to expand the use case to those who are trying to find a new place to rent. This would be fairly easy to do, and could be very valuable for those looking to move into a new rental.

Assistance

The following resources were used in the creation of major portions of this project:

- Dr. Allen's Ribbon Creation Wizard (or Ribbon Foundry)
 - Used to create the specialized ribbon for initiating the listings search
- <http://www.excely.com/excel-vba/ie-automation.shtml>
 - A piece of code from this site describing how to use VBA to do Internet Explorer searches and HTML parsing was used and heavily modified for a portion of the project.