Stubby- A StubHub Bargain Hunter

Executive Summary

Two quick facts about me led to the creation of this project:

1) I like to go out to events—sports, theatre, concerts—all those things.
2) I am an unabashed cheapskate.

And so, I wanted to design a way for me to quickly pull a list of all of the upcoming events for a city, along with the lowest price tickets and whether they were getting cheaper or more expensive—sort of a Bing Travel Advisor for event tickets. Stubby (as I lovingly call my VBA project) does exactly that—it scrapes all of the events for a particular locale off of StubHub's website, reformats it into an easy-to-use layout, and informs the user of how much ticket prices have gone up or down since their last search.

Implementation

Getting the data

I first designed a looped web query to scrape from StubHub's site, which uses the following conventions in its URL naming:

- &rows= the number of events to include on each page
- &start= the event number to start with
- &location= the city in which the event will take place
- &ae= specifies whether to include only events with available tickets
- &startDate= or &endDate= the dates to include in the search

A standard StubHub URL:

http://www.stubhub.com/search/doSearch?searchStr=tickets&searchMode=event&rows=50&start=0&nS=0&location=693;Seattle,%20Washington&ae=2&sp=Date&sd=1&startDate=2012-04-11;Next%207%20days&endDate=2012-04-17;04/17/2012

By hard-coding the query to always return 100 rows and looping the starting row to run in increments of 100, I was able to pull all the events from the given location. To identify the location, I used a VLookup table, initially only including the four locations where I am likely to attend events (the 4 S's—Sacramento, Salt Lake, San Francisco, and Seattle), although this can easily be expanded into any of the locations which StubHub includes on their site. Deleting the start and end date tags from the URL returns all events for the given location, which is my preference.

Formatting the data

My goal in designing this was to be able to sort through the ticket listings efficiently. Not meaning to sound critical, but StubHub's user interface is a little laborious, and has no back-end filtering, so I figured I would tailor the results to my preferences. I decided on the following categories, scrubbed to a user friendly-format:

- EventID- a unique identifier for each event (taken from the last seven digits of the purchase URL)
- Event Name- straight import from the query
- Date- I made the assumption that most events are at night, so just pulled the date text and formatted as a date)
- Location- Selected from a drop down list
- Lowest Price- Concatenated from the query; (this is also how I selected which rows to pull)
- Tickets Available- Concatenated from the query
- URL Link- a link in case I want to directly link to the purchasing site

Stubby Returns

	A	B	C	D	E	F	G	H	
1									
2		SELECT LOCATION:		Locations	Code				
3		Seattle		Sacramento	81690;Sacramento,%20CA				
4				Salt Lake	706;Salt+Lake+City,+Utah				
5		Stub It!		San Francisco	81;SF+Bay+Area				
6				Seattle	693;Seattle,+Washington				
7									
8	EventII	Event Name	Date	Location	LowestPrice	OldLow	Change	Tickets Available	Link
9	4038872	Blood On The Dance Floor	4/11/2012	Seattle, WA	$41.99	$41.99	0%	6	http://w
10	4025414	OFWGKTA	4/11/2012	Seattle, WA	$60.00	$60.00	0%	2	http://w
11	4069449	Mt Eden	4/12/2012	Seattle, WA	$51.99	$51.99	0%	8	http://w
12	2068159	Oakland Athletics at Seattle Mariners - Opening Da	4/13/2012	Seattle, WA	$26.00	$29.00	-10%	1,572	http://w
13	4049466	Kelly Clarkson	4/13/2012	Wenatchee, WA	$80.00	$83.00	-4%	14	http://w
14	4040443	Counting Crows	4/13/2012	Seattle, WA	$59.00	$60.00	-2%	29	http://w
15	4014660	Colorado Rapids at Seattle Sounders FC	4/14/2012	Seattle, WA	$14.00	$17.00	-18%	705	http://w
16	2068162	Oakland Athletics at Seattle Mariners	4/14/2012	Seattle, WA	$14.99	$14.99	0%	439	http://w
17	4018123	Milwaukee Mustangs at Spokane Shock	4/14/2012	Spokane, WA	$22.00	$22.00	0%	4	http://w
18	4045646	Social Distortion	4/14/2012	Kennewick, WA	$40.00	$40.00	0%	5	http://w
19	4041430	Dom Kennedy	4/14/2012	Seattle, WA	$59.00	$65.00	-9%	6	http://w
20	4030828	Kelly Clarkson	4/14/2012	Pullman, WA	$65.00	$59.00	10%	18	http://w
21	4061026	Big & Rich (Ages 21+)	4/14/2012	Tacoma, WA	$165.00	$165.00	0%	8	http://w
22	2068163	Oakland Athletics at Seattle Mariners	4/15/2012	Seattle, WA	$16.88	$16.88	0%	620	http://w
23	4047313	Chuck Ragan	4/15/2012	Seattle, WA	$63.00	$63.00	0%	10	http://w

I also built in filters so as to be able to filter for any of the categories listed above. In my design I preferred to pull in more data, and then filter on the backend, rather than filtering the query (for start and end dates, for example). An example of this:

7									
8	EventII	Event Name	Date	Location	LowestPrice	OldLow	Change	Tickets Availabl	Link
263	4065799	Les Miserables Seattle	6/27/2012	Seattle, WA	$139.00	$139.00	0%	26	http://w
264	4065800	Les Miserables Seattle	6/28/2012	Seattle, WA	$240.00	$240.00	0%	14	http://w
266	4065801	Les Miserables Seattle	6/28/2012	Seattle, WA	$200.00	$200.00	0%	6	http://w
272	4065802	Les Miserables Seattle	6/29/2012	Seattle, WA	$139.00	$139.00	0%	30	http://w
274	4065803	Les Miserables Seattle	6/30/2012	Seattle, WA	$180.00	$180.00	0%	6	http://w
277	4065804	Les Miserables Seattle	6/30/2012	Seattle, WA	$132.25	$148.00	-11%	7	http://w
280	4065225	Les Miserables Seattle	7/1/2012	Seattle, WA	$180.00	$180.00	0%	6	http://w
285	4065227	Les Miserables Seattle	7/3/2012	Seattle, WA	$139.00	$139.00	0%	24	http://w
287	4065805	Les Miserables Seattle	7/4/2012	Seattle, WA	$139.00	$139.00	0%	32	http://w
288	4065806	Les Miserables Seattle	7/5/2012	Seattle, WA	$185.00	$185.00	0%	32	http://w
289	4065807	Les Miserables Seattle	7/5/2012	Seattle, WA	$175.00	$175.00	0%	24	http://w
291	4065808	Les Miserables Seattle	7/6/2012	Seattle, WA	$242.00	$67.00	261%	8	http://w
293	4065809	Les Miserables Seattle	7/7/2012	Seattle, WA	$275.00	$275.00	0%	4	http://w
295	4065811	Les Miserables Seattle	7/8/2012	Seattle, WA	$253.00	$253.00	0%	8	http://w

I know I want to see Les Mis in Seattle, and Stubby shows me that there are a few dates which are about $50 cheaper than any of the others.

Pricing Trends

Note: I really am a cheapskate. I want to get the BEST deal possible. And so, the final thing I did was to build in a functionality that would tell me if ticket prices were going up or down. To do this, when the Stub It! button is clicked, the current data is copied to a temporary sheet. New data is imported, and the old price is matched to the new event using a VLookup. A quick calculation on the fields shows the percentage change since the last query.

Looking at the Les Mis ticket data above, we can see one date in which the lowest price more than doubled in the few minutes between when I ran these queries.

Learning Points

I have a very volatile relationship with VBA projects—when they go poorly, it's really, really bad; and when they go well I feel like I am the smartest guy on the planet. This was a good one (fortunately). One thing that I felt especially proud of was my use of nested loops, in which I would run a web query on one URL, go through the whole process of data importation and formatting, and then change URLs. (I realize that this may sound easy to experienced programmers, but for me it was a triumph!)

Another learning point from this project was the degree to which web query data needs to be manipulated in order to put it into a usable format. When we were pulling stock prices, it was pretty straightforward, but StubHub's data is much less user-friendly. For one, there are no separate tables—the entire website is one huge table. Then the information comes packaged with all sorts of extraneous information/text that needs to be trimmed down in order to be usable in the context I was proposing.

Other features that I considered including but left for future iterations of this project are calculations on the change in number of available tickets between queries (as a measure of volatility), the ability to pre-filter on dates by manipulating the web query URL (decided against this because I'd rather filter on the backend), and the integration of StubHub with other ticketing sites, such as TicketMaster or ArtTix.

A final feature that I'd like to have included but proved too time-consuming for this project's deadline is the ability to pull full ticket price data for a given event ID, rather than just the lowest price. Although I usually opt for the cheap seats, an Inception-esque layer in which the user selects the event ID and the program runs a web query based off the link provided, is pretty compelling. I plan to build out this feature in the future.

Assistance Received

I am a rock—I am an island. (I received no assistance other than the moral support of fellow-VBAers.)