

# Helium Scraper

*Scraping for any reporter, no code required*

BY LIZ LUCAS, IRE & NICAR

---

## Introduction to scraping

We've all visited websites that contain valuable information we want to analyze, such as [inspections reports](#), [lobbying expenditures](#), or a [list of jail inmates](#).

We're grateful for the [growing movement](#) among [government agencies](#) (particularly [federal](#) ones) to provide what we all want to see: the **DOWNLOAD** button.

But not everyone is there yet.

So when we can't download or copy and paste what's there, we have to give up or figure out how to literally scrape the info off the website and into a database, a spreadsheet, or something we can actually use.

Imagine tackling this site: <https://apps.sd.gov/st12odrs/LobbyistViewlist.asp?cmd=resetall>. If you wanted to end up with a spreadsheet of all registered lobbyists and their information, how would you do it?

A web scraper is simply a [piece of software](#), a [bit of code](#), or a [browser plugin](#) that does the work for us, preferably while we go have a beer.

[Helium Scraper](#) is a \$99 piece of software (for Windows only) that provides a point-and-click interface for web scraping so that you don't have to write something like this:

```
soup = BeautifulSoup(html)
results_table = soup.find('table', attrs={'class': 'resultsTable'})
output_list = []
for tr in results_table.findAll('tr'):
    row_list = []
    for td in tr.findAll('td'):
        row_list.append(td)
    output_list.append(row_list)
```

# Planning your scrape

So back to that question: how would you do it?

On [this page](#) we can search for a particular lobbyist, but to get information on ALL of the lobbyists we need to click "Show All."

**Lobbyist**

Lobbyist information and lists are available on this page.

View printable list  By Lobbyist Name or  By Employer Name for

---

*Enter a word or words to search in all columns marked below with an (\*).  
You may also choose a year to search for.  
Click the Show All link to display all Lobbyists.*

Quick Search (\*)   [Show all](#)

Year

Exact phrase  All words  Any word

We immediately get a page full of information: Year, Lobbyist Name, Address, etc. If we want to create a spreadsheet of these lobbyists, we'd probably try copying and pasting these rows into Excel. When we scroll to the bottom of the page we notice that this is one of 406 pages; there are 8120 lobbyists in total. That would be a lot of copying and pasting.

1999	DAVIS, DENNIS	2800 S VALLEYVIEW R
1999	TESSIER, DARWIN	2901 W 11TH ST

Page  of 406

Records 1 to 20 of 8120

With Helium, though, we can create a simple program to do all that automatically. First we'll make it click "Show All", then scrape all the data. Then it will click the little arrow leading to the next page, and scrape all that data. And so on, 406 times.

When you're getting ready to set up a web scraper, think through what steps you would do and then figure out how to make the computer do those things.

## Helium terminology: Kinds, Actions and Data

Helium Scraper helps you build a web scraper using a point-and-click interface that operates with two general concepts: kinds and actions. **Kinds** are the pieces of the website you'll need the scraper to act on. Think of them as building blocks. **Actions** are what we create to arrange those building blocks in a particular order.

For example, we need the scraper to click "Show All" before it can scrape any data. First we have to tell the scraper how to recognize the "show all" button by creating a **kind**, and we tell it to actually click the button by creating an **action**.

Additionally, we want to create a **kind** for each column of data we want: year, name, address, etc. That way the scraper can recognize the pieces of data we want. The **action** we'll create is to take those **kinds** and extract them to a table.

The result of the scrape is the **data**, which Helium Scraper stores in a database that you can query. You can also export the **data** as a Microsoft Access database.

These are the basic steps, but Helium Scraper has a lot of additional functionality. For example, you can make the program wait between each click, which is helpful because certain web browsers or web pages won't allow you to make too many requests too quickly. You can also send the results of your scrape to different tables if you want to create a relational database.

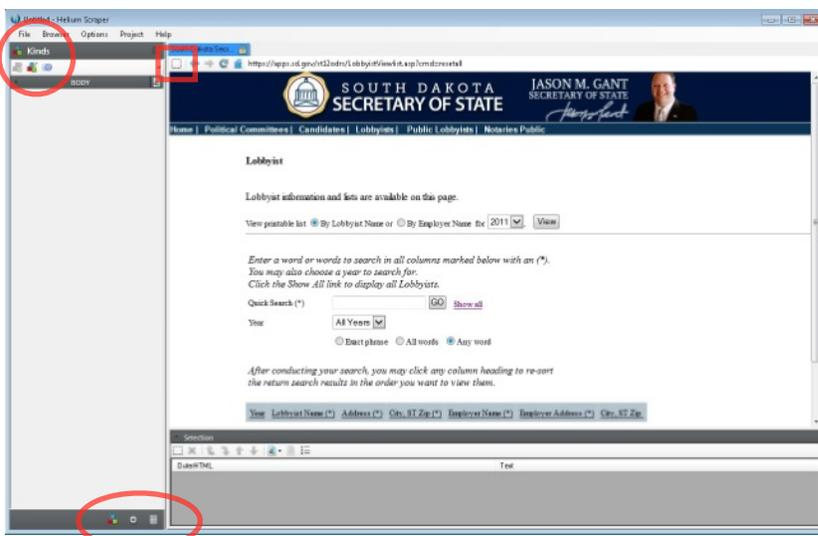
## Get acquainted with the Helium Scraper window

When you open Helium Scraper, you'll notice that there's a browser inside the program, where you should navigate to the page where you want to start scraping. (Get as close to the data as you can so you don't add extra work by having to click through a bunch of pages to get to the data.)

The Select button allows you to select elements on a webpage to create a **kind**. When Select is off, you can navigate around the webpage as you would in a normal browser.

The left sidebar has the header **Kinds**; here we create the **kinds** we want to work with.

At the bottom of the sidebar are three buttons. When you hover over them, you'll see "kinds", "actions" and "database." You'll use these buttons to flip between the three steps of the scraper. Currently we are working with **kinds**.

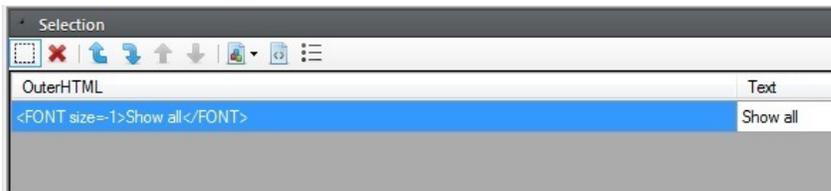


## Building the scraper

We'll create our first kind by telling the program how to recognize the "Show All" button. Make sure the "Select" option is activated: 

When you hover over elements on the web page, they will temporarily turn purple if Select is activated.

Hover over "Show All" and click. In the bottom pane an element has been added:



Over in the left sidebar, click the button "Create kind from selection."



This will look at all the shared properties of what you selected. Selecting only one element (as we did) will create a very specific kind that will only recognize that one element. In this case that is OK.

Enter the name "ShowAll." Now we have a new kind listed in the sidebar below BODY, along with all the properties of that kind.

Now we'll create kinds for each of the data elements we want to scrape. In the browser window, click the Select button (to turn it off) and then click Show All so that the lobbyists' data appears.

Start with the first column: Year. Turn Selection back on and select the first three or four values in the Year column. To select more than one thing, hold Ctrl down while you click. **Be sure to click on the text, and not the whitespace in the cell.**

Year	Lobbyist Name *
2011	KORT, MARCELO
1999	HUETTL, DARLENE
1999	NELSON, DAVID
1999	SCHROYER, CHUCK
1999	WILLIAMS, H. WAYNE
	HAUSCHIND

Now go back to the left sidebar and, under Kinds, click "Create kind from selection" again. Name it "Year." We now have two new kinds listed in the sidebar, along with their properties: "ShowAll" and "Year."

Repeat this process to create a kind for each column in the table:

"Lobbyist", "Address", "CityStZip", "Employer," "E\_Address", and "E\_CityStZip."

*If you get the message "These elements seem to be completely different kind of elements because they have different tag name" be sure you're clicking the **text**, not the **whitespace**, in each cell.*

Hint: you can expand and collapse each kind by double-clicking on the gray bar.

You can check to make sure that any of the kinds will scrape what we want by clicking the button: “Select kind in browser.”



This will highlight in purple all the elements on the current page that match this kind. In this case, you should see every value in the “Year” column on our webpage highlighted.

For this scrape, we’re almost done creating all the kinds we need. We still need to create a kind that Helium can use to identify the “next” button that pages through all the lobbyists. With Selection turned on, click the arrow that brings you to the next page of lobbyists:



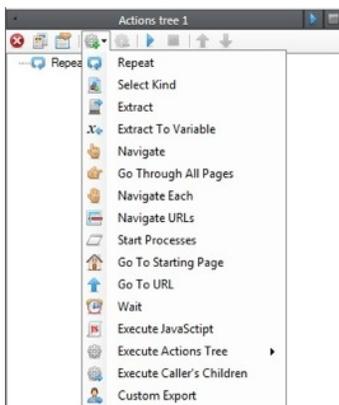
Create a kind from this and call it “Next.” In order to make sure that this kind recognizes the button on every page, we’ll need to add to this kind. Turn off Selection and navigate to the next page of data. Then turn Selection back on and click the *next* button on this page as well.



Now we’ve got all the necessary kinds created, we can turn to actions. At the bottom of the sidebar, click the actions button: 

“Action tree 1” already exists, so go ahead and expand it by double-clicking on the bar. So far it only has one action: “Repeat 1 times”.

There are a number of actions you can add to an action tree:



The ones we’ll use here are “Repeat”, “Extract”, and “Navigate.”

“Repeat” will tell Helium how many times we want to execute the following commands.

“Extract” will actually pull the data from the page into a table.

“Navigate” tells the scraper to click on items on the webpage (example: “Show All”).

The first **action** we want Helium to take is to click the “Show All” button. So we’ll clicking the “New action” button  and select “Navigate.”

In the dialogue box that pops up, the option “Select kind” should already be checked and “Show all” appears in the drop-down menu. Just hit “OK.”

Now, under “Repeat 1 times” in our action tree appears “Navigate: ShowAll.”

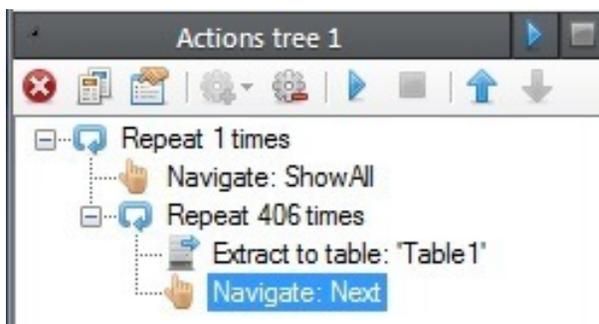
After our scraper completes this **action**, what do we want it to do next? We want it to scrape all the data and then go to the next page and repeat. We want it to do this 406 times so that it captures all of the pages.

Add a new **action**, this time “Repeat.” In the dialogue box, enter “406” in the space for “Iterations.” Hit “OK.” Now “Repeat 406 times” appears under our last “Navigate” command. Click on “Repeat 406 times” and add another **action**: “Extract.”

In the box that pops up, choose all the elements we want extracted into our data table; everything but “ShowAll” and “Next.” Hit OK. Another box will pop up showing you the layout and some details of your table, automatically named “Table 1.” Change the table name if you like, but leave the rest of the defaults. Hit OK.

Add another **action**, this time “Navigate” again. In the drop-down menu, choose “Next” and hit OK.

Your action tree should look like this:



Click the run button  ! It will take a few minutes. This is where you get up and stretch.

## Final results

When it’s finished, click the **database** button at the bottom of the sidebar:



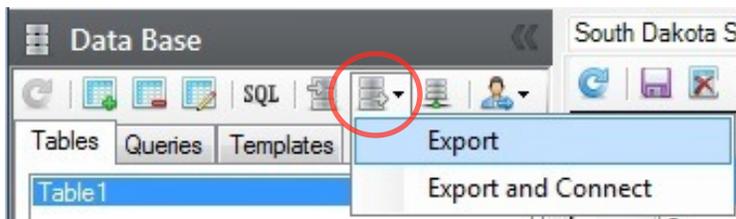
You should see your “Table1” or whatever you named it. Double-click it and your data should appear in the main window. Check to make sure all 8120 records are there.

You can query the data with SQL right in the program, or you can export the data as a Microsoft Access database (.mdb) or as a CSV (comma-delimited text file).

To query the data here, open a new query with the SQL button in the sidebar:



To export as an Access database, choose Export in the sidebar:



To export as a CSV, go to the main window:



## Final tips:

It's a good idea to test certain parts of your scraper as you go. If you're building something that has a lot of steps, make sure each step works before moving on to the next thing.

Save multiple copies (.hsp files) as you go so that you can easily backtrack.

If the scraper doesn't seem to be accurately picking up an element, try clicking more examples. Sometimes exactly *where* you click makes a difference: for example, clicking on the text in a cell as opposed to the whitespace in the same cell.