VALE Walkthrough: Chrome Web Scraper

Original by Damian Gordon

Chrome webscraper, and tools like it, allow you to automatically scrape data from websites. You specify the path you would like the program to take as it runs through website, and which information to scrape. It outputs the data in a convenient csv file format.

This tutorial will teach you how to use chrome web scraper, by creating a sitemap for a forum for people who hear voices

**Required**: Chrome Web Scraper, Internet Connection and [FORUM URL: <http://hvn.forumatic.com/>].

**Part A - Getting Started**

1. Open Google Chrome and click on Chrome Web Store
2. Search for “Scraper” in extensions
3. The first search result is the “Scraper” extension
4. Click the ‘add’ to chrome button.

**Part B – Data Harvesting**

**Creating a New Site Map**

1. Navigate to the following URL: <http://hvn.forumatic.com/>
2. Right click anywhere on the screen and click ‘Inspect Element’
3. Click on the Web Scraper tab on the top right of the window that appears
4. Click on ‘Create new sitemap’ and select ‘create new sitemap’ from the drop down list
5. Type in an appropriate name for the sitemap of the forum, and paste the URL: <http://hvn.forumatic.com/> into the start URL form.
6. Click ‘Create Sitemap’

**Creating the Site Map**

Selectors tell the web scraper which links to follow, and which data to harvest. There are a number of different types of selectors. The most important of these are:

* Link: Link selectors tell the web scraper which links to open from a certain page
* Pop-up Link: Same as link selectors, but for following links which automatically open in a new window
* Text: Highlight text data that you wish to scrape.
* Element: Select elements within the webpage. A way to group certain data together. What elements are and how to use them will be made clear later in the tutorial
* Image: Select images

When creating selectors, one needs to specify the selectors which are it’s ‘parents’. The parent of a selector is the selector that will be executed before the child selector. ‘\_root’ is the first directory of the start URL you specified when you created the sitemap, and all selectors will have \_root as their ultimate parent.

1. On the homepage of the Hearing Voices Network forum, we can see there are 3 sub-forums. We don’t care about the posts in the ‘Introductions’ forum, but we would like to scrape all posts from both ‘Peer Support’ and ‘The Think Tank’. In order to do this we need a selector to follow both of these links.
2. Click ‘Add new selector’
3. Type ‘subforum’ for the selector id.
4. As we want to specify links for the web scraper to follow, set the type to ‘Link’.
5. Because there are multiple links (2 subforums) on the page we are on now that we want to follow, check the box that says ‘Multiple’.
6. Under ‘Selector’ heading, click ‘Select’. Note that if you now simply click on the links to one of the subforums, chrome will open those links like normal. To avoid this, click ‘enable key events’. This allows you to select items with your keyboard.



1. Hover your mouse over the ‘Peer Support’ subforum link, highlighting it in green, and press the s key. Then, do the same for ‘The Think Tank’. Now, both of these links should be highlighted in red. Click ‘Done selecting!’
2. The parent selector is already set to \_root, and as this is the first thing we want the web scraper to do, we do not need to change this. Click ‘Save Selector’.
3. Once web scraper has opened the subforum links, we would like to open all the threads within each subforum. Open up the ‘Peer Support’ subforum. The first thing we should notice is that there are multiple pages in this subforum (and the ‘Think Tank’), each with multiple threads. When we have long lists that go on for multiple pages like this, it is called pagination. So the first thing we need to do is tell the web scraper to open every page of each subforum.
4. Click ‘Add new selector’. Give it id ‘pagination’, and selector type ‘Link’.
5. If you scroll to the bottom of the page, you will see a link called ‘Next >’. Set this link as the selector the same way that you did before. As there is only one link for this selector to follow, leave the ‘multiple’ box unchecked.
6. We want web scraper to follow the ‘Next>’ link when it opens to the subforum, so we set the parent selector to be subforum.
7. However, we don’t want the ‘Next’ link to be followed only once, opening the second page of the subforum and then stopping, we want it to open every page of the subforum. Basically, we want to call this selector on the page that this same selector opens. This is called recursion. To do this, we need to set the ‘pagination’ selector as a parent of itself. One can set multiple parents of the same selector by holding down ctrl while clicking to select parents.
8. Once both ‘subforum’ and ‘pagination’ have been set as parents of ‘pagination’, click ‘Save selector’.
9. Now we need a selector to open all the threads in both subforums. Create a new selector called ‘thread’, of type ‘Link’.
10. Remembering to enable ‘key-events’ start selecting threads on the first page of the ‘Peer Support’ subforum. Notice that after 2 or 3 threads have been selected, web scraper picks up the pattern and highlights all the threads. Scroll down to make sure it has got all of them - sometimes the pattern it finds does not include all links you wish to follow. If some threads are not highlighted, then select those as well. When you’re done, click ‘Done selecting!’.
11. You want to select all threads in the subforums, as well as all the links followed by the pagination selector. So set both ‘subforum’ and ‘pagination’ as parents of this selector.
12. Open one of the threads in the ‘Peer Support’ subforum. We want to save every post from this thread, along with the date and author of the post. If we simply made three selectors for each of these fields, and set ‘thread’ as their parents, then web scraper would put each of these pieces of information in a different row, when we really want the date, author and corresponding post body all in the same row.
13. In order to do this, we first need to create an element selector. Create a new selector and name it ‘post\_element’, and set the selector type to ‘element’.
14. As before when selecting threads, select each post on the page including information about the author, title and post body. An example of a single element is shown in the picture below. Make sure everything shown in the picture is highlighted before you select is as an element. Choose ‘thread’ as the parent selector.



1. Now we can make three separate selectors for author, date, post\_body. Doing this will be easier if, in the webscraper window, you first navigate into the post\_element selector, as shown in the picture below



1. Create the three selectors for author, date and post body. In this case, selector type is ‘Text’, and parent selector is post\_element. Note that because there is only one post body, author and date per element, you do not tick the ‘multiple’ box, and only have to select one example of each type, even if there are many posts in the thread you are using.
2. You are almost done, but now you notice that some threads actually go on for multiple pages!
3. In order to deal with this, you need to create a pagination selector just as we did earlier, this time for finding all pages within a thread.
4. Make this selector, remembering to set the appropriate parents.
5. Your new within-thread pagination selector should now find all the pages within a thread. However, the post\_element selector you made previously does not run after this new selector, and so you won’t get any of the posts after the first page. Therefore, you need to edit the post\_element selector you made earlier. Navigate to it and click ‘Edit’.
6. All you need to do is add within-thread-pagination as a parent of this selector, at the same time keeping ‘thread’ as a parent selector as well.
7. At this point the site-map should be complete. Before you start scraping you will want to review it to make sure it is working properly. Navigate to the selector graph, as shown in the picture below:



1. This will give you a graph showing the map you have created. Click on a node to reveal all the children of that selector. A picture of what your site map should look like at this point is shown in the picture below:



Note that because the pagination and within\_thread\_pagination are recursive, you can keep clicking on them and the map will never end.

**Downloading the Data**

1. If the site map looks good, we can start scraping the website. Click on ‘Scrape’, found lower down in the list where you found the selector graph.
2. The request interval and page load delay options are there because sites will block your access if they notice a bot is automatically browsing their site. By adding a delay to how quickly you move through the site, you can usually get around this. If you try and scrape a website and it doesn’t work, experiment with using longer request intervals and page load delays.
3. For this website, the default values for request interval and load delay are fine. Click ‘Start scraping’. A new window will open up showing you how Webscraper is navigating through the site. The window will close automatically when it is finished scraping the entire site. If you close the window manually before this point, scraping will be stopped, but the data collected up until that point will still be available,
4. In the same menu where you found the selector graph and scrape options, navigate to ‘Browse’. This will show you what your data looks like. If it looks good, In the very same menu, click on “Export data as CSV”. You can then click ‘download now!’ to get download your data in CSV format.