Selenium How To Configure

? Unknown Attachment

Step 1. Launch Aspire and open the Content Source Management Page

Launch Aspire (if it's not already running). See:

- Launch Control
- Browse to: http://localhost:50505. For details on using the Aspire Content Source Management page, please refer to Admin UI

? Unknown Attachment

Step 2. Add a new Selenium Content Source

To specify exactly what shared folder to crawl, we will need to create a new "Content Source".

To create a new content source:

1. From the Content Source , click on "Add Source" button.
2. Click on "Selenium Connector".

? Unknown Attachment

Step 2a. Specify Basic Information

In the "General" tab in the Content Source Configuration window, specify basic information for the content source:

1. Enter a content source name in the "Name" field.
   a. This is any useful name which you decide is a good name for the source. It will be displayed in the content source page, in error messages, etc.
2. Click on the **Scheduled** pulldown list and select one of the following: *Manually, Periodically*, *Daily, Weekly or Advanced.*
   a. Aspire can automatically schedule content sources to be crawled on a set schedule, such as once a day, several times a week, or periodically (every N minutes or hours).For the purposes of this tutorial, you may want to select Manually and then set up a regular crawling schedule later.
3. Click on the **Action** pulldown list to select one of the following: *Start, Stop*, *Pause,* or *Resume.*
   a. This is the action that will be performed for that specific schedule.
4. Click on the **Crawl** pulldown list and select one of the following: *Incremental, Full*, *Real Time,* or *Cache Groups.*
   a. This will be the type of crawl to execute for that specific schedule.

After selecting a Scheduled, specify the details, if applicable:

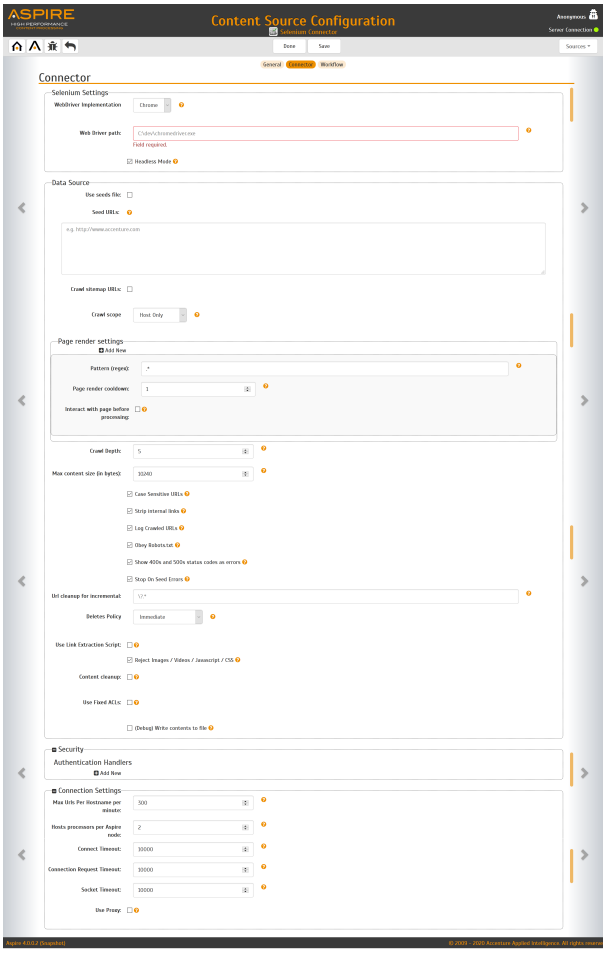
- *Manually*: No additional options.
- *Periodically*: Specify the "Run every:" options by entering the number of "hours" and "minutes."
- *Daily*: Specify the "Start time:" by clicking on the hours and minutes drop-down lists and selecting options.
- *Weekly*: Specify the "Start time:" by clicking on the hours and minutes drop-down lists and selecting options, then clicking on the day checkboxes to specify days of the week to run the crawl.
- *Advanced*: Enter a custom CRON Expression (e.g. 0 0 0 ? * *)

You can add more schedules by clicking in the **Add New** option, and rearrange the order of the schedules.

If you want to disable the content source just unselect the the "Enable" checkbox. This is useful if the folder will be under maintenance and no crawls are wanted during that period of time.

Real Time and Cache Groups crawl will be available depending of the connector.

Step 2b. Specify the Connector Information



In the "Connector" tab, specify the connection information to crawl the Selenium.

1. Select the Web driver implementation you want to use, this is related to the browser that will be controlled by Selenium.
   Currently there are two options
   a. Chrome.
   b. Firefox.
2. Set the path to the web driver.
   Once a driver implementation is selected, a path to the executable driver must be provided.
3. Set the headless mode if required, if the connector is running on headless mode UI window will be displayed for the browser.
4. Select the "Use seeds file" if the connector should read the seed urls from a file.
   a. If selected, set the path to the seeds file.
   b. Otherwise fill the "Seeds URLs" text area with the desired urls.
5. Check the "Crawl Sitemap urls" if there are sitemaps that must be processed by the crawler.
   a. If selected, select the "Use sitemaps from file" if the urls are on a file.
      i. Select the sitemap urls file.
   b. Otherwise set the "Sitemap URLs" text area with the desired sitemaps.
6. Select a Crawl Scope:
   a. Host Only: The crawler will not process any url that has a different hostname from the seed urls.
   b. Custom: The crawler will process any url that matches the provided regular expression.
   c. Everything: The crawler will process any url it finds.
7. Page Render Settings, this section will tell the crawler how to handle the urls rendered by the browser.
   This section allows for multiple entries, each entry must specify:
   a. Pattern: Regular expression that will be evaluated against each url, if it matches the remaining elements will apply.
   b. Page render cooldown: Time (in seconds) the crawler will wait after the page is done loading before retrieving the content.
   c. Interact with page before processing:
      i. If selected, provide the "Interaction Script" with the instructions for the selenium driver before retrieving the content.
8. Set the crawl depth
   a. The max number of "jumps" the crawler goes through a seed url to reach the current url, leaving 0 will make the crawler to process only the seed urls.
9. Set the max content size (in bytes)
   a. The crawler will not process any url that contains more than the allowed size.
10. Set case sensitive urls
    a. If selected, the crawler wll treat the urls as case sensitive.
11. Set strip internal links
    a. If selected, the crawler will remove internal links before processing the urls.
12. Set obey robots.txt
    a. If selected, the crawler will check with the site robots.txt and page meta tags if the url should be processed.
13. Set show 400s and 500s status as errors
    a. If selected, the crawler will report as error any incoming response with 4XX and 5XX status code.
14. Set url cleanup for incremental regular expression
    a. This expression will be used if the urls must be cleaned up in order to match the previous crawl urls.
15. Set deletes policy:
    a. Inmmediate: the deleted urls are reported in the crawl they were first discovered.
    b. Time based: the deleted urls will be reported on an incremental crawl after X days have passed.
    c. After X incrementals: the deleted urls will be reported on an incremental crawl after X incremental crawls have been executed since they were first discovered.
16. Set link extraction script flag.

a. If selected, provide a script with the steps to retrieve the discovered urls from the current page. Please refer to the Selenium FAQs for more details.

17. Set Reject Images / Videos / Javascript / CSS
    a. If selected, multi-media, javascript and css urls will not be processed by the crawler.
18. Set content cleanup
    a. If selected, set the following values:
        i. URL pattern: the regular expression that will be evaluated against each url.
        ii. CSS classes to remove: Comma separated list of every css class that should be removed from the url content.
        iii. Cleanup pattern: All the elements in the url content matching regular expression will be removed.
        iv. Content type pattern: Regular expression to be evaluated against the url mime type, if it matches the cleanup will be applied
19. Set Fixed ACLs:
    a. If selected, provide the following values for group and user ACLs:
        i. Domain.
        ii. Name.
        iii. Type (Allow, Deny)
20. Set Write contents to file.
    a. If selected, the raw content of the url will be dumped into a local file.
21. Set the Authentication Handlers.
    a. Host: Hostname of the urls that will apply this handler, if no hostname is set, it will be used for all.
    b. Port: Port of the url, if set to -1, any port will be accepted
    c. Login Url: Url to the login page
    d. Realm: User realm
    e. Domain: User domain
    f. User: User name
    g. Password: User password
    h. Authentication implementation: The crawler will use this configuration to log in the page.
        i. Simple Authentication:
            1. Username field selector type:
                a. Class
                b. CSS
                c. Id
                d. Name
                e. X Path
            2. Username field: Field on the login form where the username should be set
            3. Password field selector type:
                a. Class
                b. CSS
                c. Id
                d. Name
                e. X Path
            4. Password field: Field on the login form where the password should be set.
            5. Custom field:
                a. Field type:
                    i. Class
                    ii. CSS
                    iii. Id
                    iv. Name
                    v. X Path
                b. Field name: Field on the login form where the value should be set.
                c. Field value: Value of the custom field.
            6. Submit field selector type:
                a. Class
                b. CSS
                c. Id
                d. Name
                e. X Path
            7. Submit field: Field on the login form where the submit button is located.
        ii. Scripted Authentication:
            1. Authentication Script: Groovy script with all the instructions to login in the site. Please refer to the Selenium FAQs for more details
    i. Verification implementation: The crawler will use this configuration to check if the session on the page is still valid.
        i. Simple verification: All the specified fields must be present on the page, otherwise the session will be considered as terminated
            1. Field selector type:
                a. Class
                b. CSS
                c. Id
                d. Name
                e. X Path
            2. Field name: Name of the field to be checked.
        ii. Scripted verification
            1. Verification Script: Groovy script with the process to check if the session is still active, must return true or false. Please refer to the Selenium FAQs for more details.
    j. Max Urls Per Hostname per minute: Max urls to be process in a minute for a single host.
    k. Hosts processors per aspire node: Each hosts processor processes queued URIs for only one host at the time. Only one Aspire node in the cluster will activate a processor for a given host at the time, ensuring the throttle limits across the entire cluster.
    l. Connect timeout: Timeout in milliseconds to establish a connection.
    m. Connection request timeout: Timeout in milliseconds used when requesting a connection from the connection manager.
    n. Socket timeout: Timeout in milliseconds for waiting for data or maximum period inactivity between two consecutive data packets.

o. Use proxy:
    i. If selected, set the following:
        1. Proxy host.
        2. Proxy port
        3. Use credentials:
            a. Proxy User
            b. Proxy Password

Step 2c. Specify Workflow Information

? Unknown Attachment

In the "Workflow" tab, specify the workflow steps for the jobs that come out of the crawl. Drag and drop rules to determine which steps should an item follow after being crawled. This rules could be where to publish the document or transformations needed on the data before sending it to a search engine. See Workflow for more information.

1. For the purpose of this tutorial, drag and drop the *Publish To File* rule found under the *Publishers* tab to the **onPublish** Workflow tree.
    a. Specify a *Name* and *Description* for the Publisher.
    b. Click *Add*.

After completing this steps click on the **Save** then **Done** and you'll be sent back to the Home Page.

Step 3: Initiate a Full Crawl

Now that the content source is set up, the crawl can be initiated.

1. Click on the crawl type option to set it as "Full" (is set as "Incremental" by default and the first time it'll work like a full crawl. After the first crawl, set it to "Incremental" to crawl for any changes done in the repository).
2. Click on the Start button.

During the Crawl

During the crawl, you can do the following:

- Click on the "Refresh" button on the Content Sources page to view the latest status of the crawl.
  The status will show **RUNNING** while the crawl is going, and **CRAWLED** when it is finished.
- Click on "Complete" to view the number of documents crawled so far, the number of documents submitted, and the number of documents with errors.

If there are errors, you will get a clickable "Error" flag that will take you to a detailed error message page.

Step 4: Initiate an Incremental Crawl

If you only want to process content updates from the Selenium (documents which are added, modified, or removed), then click on the "Incremental" button instead of the "Full" button. The Selenium connector will automatically identify only changes which have occurred since the last crawl.

If this is the first time that the connector has crawled, the action of the "Incremental" button depends on the exact method of *change* discovery. It may perform the same action as a "Full" crawl crawling everything, or it may not crawl anything. Thereafter, the Incremental button will only crawl updates.

Statistics are reset for every crawl.