



Web Crawler Documentation

[Introduction](#)
[SimSage Source configuration](#)
[The Web Crawler Tab](#)
[the Metadata Tab](#)
 [Metadata Transforms](#)
[The ACLs Tab](#)
[The MimeType Tab](#)
[The Schedule Tab](#)
[External Crawler Configuration](#)
[Troubleshooting](#)
 1. Error Messages in Crawler List
 2. Testing Platform Connectivity
 3. Reviewing Crawler Logs

Introduction

This Document sets out the steps need to step up different types of crawlers on the SimSage platform. Each crawler will have its own unique “source” configuration. It is vital that you do not set up the same source twice (with the same parameters). Setting up the same source twice will stop both sources working until one of the is removed.

SimSage Source configuration

1. Select your Organisation (SimSage shown in the image below)
2. Select your knowledge-base from the drop-down box (demo knowledge-base shown below)



Organization	Enabled
SimSage	yes

3. Click on Document Management link in the left hand side of the menu
4. Click on the + Add Source button

5. In the General tab

- set the Crawler Type to “Web Crawler” (for instance)
- give the Crawler a Name (test web crawler in the image below)
- Select the “Web Crawler” tab to set up your web crawler

The Web Crawler Tab

- Navigate to the “Web Crawler” tab.
- Populate the base url field.
- Populate the exclude css csv field. As a general rule we recommend using “header, footer” to exclude any repeating headers or footers from your HTML.

Not all webpages use semantic HTML tags such as <header> or <footer>. Sometimes websites use regular div tags with class names. For example, <div classname="footer">. In this case you should enter div.footer instead of “footer”.

Add Source

General **Web Crawler** **Metadata** **ACLs** **Check Document** **Schedule**

http/s or file:// base url (non-file paths must end with /) *
(e.g. <https://simsage.ai>)

Pre-render Javascript websites

Valid extensions (html is always valid)
Ignore extensions (html cannot be ignored)

User-agent (leave blank for default)
web-crawler's user-agent

Username
optional basic auth username

Password (leave blank to keep previous)
basic auth password

use Cookie Authentication
Cookie Authentication endpoint cookie authentication endpoint

Include css csv
css/html root fragments to include csv

Exclude css csv
css/html root fragments to exclude csv (e.g. header, footer, div.class-name)

csv include words
csv words, include articles by words [optional]

csv exclude words
csv words, exclude articles by words [optional]

csv allowed domains
csv prefix list of other domains to crawl (e.g. https://drive.google.com) [optional]

exclude paths (csv list)
a set of possible paths values, separated by commas (csv) to exclude pages (e.g. /images/) [optional]

openid-configuration endpoint
openid-configuration endpoint (e.g. https://login.microsoftonline.com/<tenant-id>/well-known/openid-configuration)

OIDC/OAuth client id

OIDC/OAuth secret

Impersonation user
the user to impersonate for Google drive (e.g. john@abc.com)

Google drive json key
the Google drive json key identifying the service account to use to access and impersonate user-drive data. Leave empty if you've already set this value previously and don't want to change it.

Close **Save**

field	meaning and content
http/s or file:// base url	a compulsory field with a comma separated list of http:// https:// or file:// URLs / URIs to visit. This set forms the seed set for the initial crawler.
pre-render Javascript websites	a special flag that uses a browser to fetch and render remote pages that use active javascript
Valid extensions (html is always valid)	A comma separated list with valid file extensions (e.g. doc, docx, pdf). All file extensions are fetched if this field is left empty. Leave this field empty if you are planning to use "Ignore extensions".

Ignore extensions (html cannot be ignored)	A comma separated list of extensions that must not be included. Leave this field empty if you are planning to use “Valid extensions”.
User-agent	a string for the user-agent to pass to remote sites in your request headers. Leave empty for default. Default is “Apache-HttpClient/UNAVAILABLE”
Username	the BasicAuth username to send if set along with password in the request headers.
Password	the BasicAuth password to send in the request headers if set.
use Cookie Authentication	A special mode where the crawler uses a Cookie Authentication endpoint to fetch an auth cookie using the Username and Password above for authentication
Cookie Authentication endpoint	Use when Cookie Authentication is enabled. The web crawler authenticates using this URL and the Username / Password above using a form post using “uid” and “key” form parameters.
Include css csv	a comma separated list of CSS selectors for elements to keep in an HTML document. All elements are kept if this field is left blank. Keep this field blank if you’re using the “Exclude css csv” (see: J Use CSS selectors to find elements)
Exclude css csv	a comma separated list of CSS selectors for elements to exclude from an HTML document. Keep this field blank if you’re using the “Include css csv”. (see: J Use CSS selectors to find elements)
csv include words	A comma separated list of words used to filter HTML pages. Any literal “word” found inside the text of an HTML page qualifies this page to be crawled / used by SimSage (e.g. accounting, invoice). All pages are included if left blank. Keep this field blank if you intend to use the “csv exclude words” filter.
csv exclude words	A comma separated list of words used to filter HTML pages. Any literal “word” found inside the text of an HTML page eliminates this page from being crawled / used by SimSage. No pages are excluded if left blank. Leave this field blank if you intent to use the “csv include words” filter.
csv allowed domains	A comma separated list of URL prefixes (e.g. https://another.server.com/folder1/) of other

	domains apart from the seeds above to allow crawling across.
exclude paths	A comma separated list of path fragments used to excluded pages from being crawled (e.g. /invoices/)

Extra security headers for Bearer Authentication

field	meaning and content
openid configuration endpoint	Used for OpenID authentication. The crawler will use this endpoint to get the OpenID configuration required for fetching JWT tokens for Bearer authentication. (e.g. https://server.com/.well-known/openid-configuration)
OIDC/OAuth client id	Used for OpenID authentication. This is the client ID passed to the identification endpoint for acquiring a JWT token.
OIDC/OAuth secret	Used for OpenID autentication. This is the client secret passed to the identification endpoint for acquiring a JWT token.

Google drive authentication

field	meaning and content
impersonation user	A users (an email address) to impersonate to access Google drive links on web pages.
Google drive json key	A Google drive json key (see Google drive setup) for authenticating against Google drive for reading, converting and downloading Google drive links seen on web pages.

NB. The Google drive impersonation user might not have access to links of personal drives of other users if this impersonation user is not in the ACLs for that link. This is a Google drive sharing feature. All shared drive links can be read as long as the drive has been shared with the impersonation user.

the Metadata Tab

The web crawler has the option to replace certain fields of a SimSage record with metadata from HTML pages.

NB. You must save this crawler before altering the mappings.

SimSage has a series of special names that can be used to map the main items used in SimSage. You can map any item, even if SimSage doesn't directly map it to one of its main items. Items that don't map are just mapped as metadata and can be found through metadata searches.

The main items understood by SimSage are

SimSage item	Description
url	the primary key of a SimSage item. If the primary key is an http or https reference the item can be directly access from the remote system.
title	The title of a SimSage record, important in search as it gives each record a score boost
body	The main content of a SimSage record, the default text that is searchable
author	The author of a SimSage record
created	The created Date-time (must be a date as a minimum, various formats supported) of a SimSage record
last-modified	The last-modified Date-time (must be a date as a minimum, various formats supported) of a SimSage record

Below is an example of SimSage mapping HTML meta items called “og:description”, “og:title”, and “og:url” into its body, title, and url fields respectively. Note that the type of the asset will be changed from HTML to TEXT if the “body” text is no longer HTML. This applies to both “<meta name=...>” and “<meta property=...>” data found in your document.

Edit Source: gdrive

General Google-Drive Crawler **Metadata** ACLs Check Document Schedule

+ Add Metadata Mapping

UI Display-name (optional)	SimSage Metadata name	Source metadata name	Delete
created	created	created	Delete
last modified	last-modified	last-modified	Delete
document type	document-type	document-type	Delete
Ui display-name	body	og:description	Delete
Ui display-name	title	og:title	Delete
Ui display-name	url	og:url	Delete

↳ Metadata Transform (optional)

Metadata Source: none Metadata Destination: none Matching Regular Expression: regex, e.g. 'http(s)?://domain.com/(?<param>.)*/'

Close Test Reset Delta Save

Any other metadata fields (whether included or not) are mapped into the metadata of each SimSage record.

Metadata Transforms

SimSage has the ability to transform / enhance metadata using regular expression matching. Imagine there was an important ID number part of your URL that you want to insert into every title of your document.

This is what metadata transforms are for.

Metadata transform field	Description
Metadata Source	what metadata field to apply a regular expression to for processing / extracting information from.
Metadata Destination	if successful, what metadata field to apply the new data to
Action	How to apply the new data, one of {after, before, replace}. After places the result after the existing field's metadata. Before places the result before the existing field's metadata. Replace 'replaces' any data in the destination metadata field.
the Text	What text to create. This can be any text, including <name> (diamond bracket name) fields as used in the regular expression.
Matching Regular Expression	the regular expression to apply to each metadata source field.

Here is an example. We have an ID number that doesn't occur in the title of our documents but the URL of our documents. We would like to add this ID with a "Customer ID" prefix to each title of matching documents. The ID number only occurs in certain URLs of our "[orders.com](#)" website as part of the second path entry of an order URL (e.g. <https://orders.com/customer/123>). We set up the values as shown below.

Edit Source: gdrive

General	Google-Drive Crawler	Metadata	ACLs	Check Document	Schedule
+ Add Metadata Mapping					
UI Display-name (optional)	SimSage Metadata name	Source metadata name			
created	created	created	Delete		
last modified	last-modified	last-modified	Delete		
document type	document-type	document-type	Delete		
↳ Metadata Transform (optional)					
Metadata Source	Metadata Destination	Matching Regular Expression			
url / id	title	http(s)?://orders.com/v/customer/(?<id>.*\$)			
Action	the Text (including regex <group> names in diamond brackets)				
before	Customer ID <id>				
Close Test Reset Delta Save					

Any data ingested by this source will now change the title of each record that matches the regular expression applied to its URL field. Records that do not match the regular expression are not modified. Take special care with expressions like this to allow for ending "/" extra slashes where applicable.

Changing the Metadata Transform does require any existing data (already ingested data) to be re-processed.

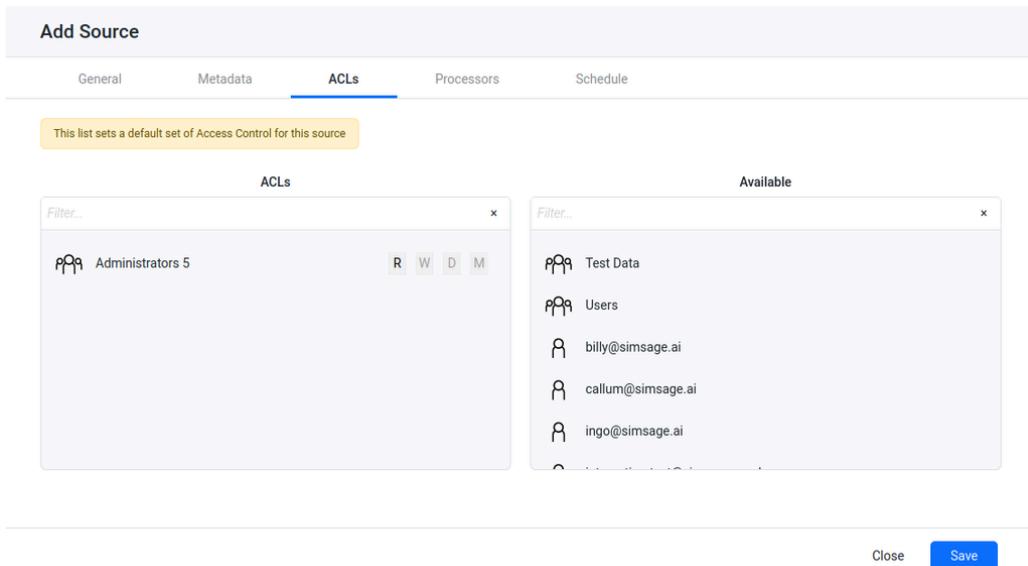
The ACLs Tab

In this tab we can customise the security based on existing SimSage Users and Groups.

NB: Users and Groups are automatically imported from external systems as they are crawled, where available.

In the below screenshot example we have selected the SimSage default User group as having access. Note, if you have opted to use the “allow anonymous access to these files” in the General source configuration, these ACLs settings will be overridden.

SimSage always recommends setting up a group initially in case you change your mind at any point.



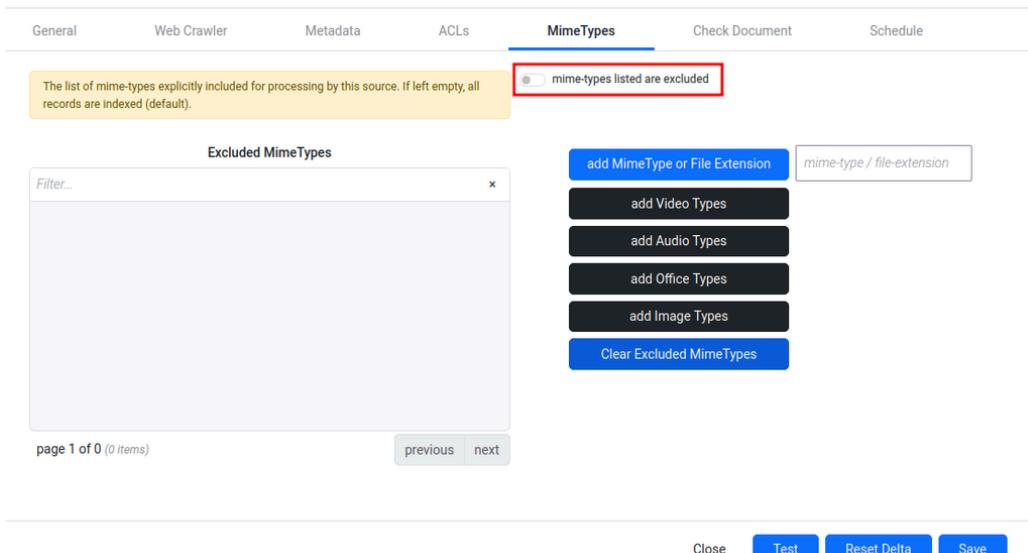
This screenshot shows the 'Add Source' interface with the 'ACLs' tab selected. A yellow box highlights a note: 'This list sets a default set of Access Control for this source'. The 'ACLs' section contains a list of users and groups, with one item selected: 'Administrators 5' (with R, W, D, M permissions). The 'Available' section lists several users and groups: 'Test Data', 'Users', 'billy@simsage.ai', 'callum@simsage.ai', and 'ingo@simsage.ai'. At the bottom right are 'Close' and 'Save' buttons.

The MimeType Tab

MimeTypes are the Internet equivalent of file-types. There are two modes in SimSage for gathering data. The default mode is to accept all data seen by a source/crawler and index/process that data.

It is however possible to just have SimSage take stock about what is contained inside a remote system without doing any more work.

Here you can set how you want particular types of files to be processed for a particular source.



This screenshot shows the 'Add Source' interface with the 'MimeTypes' tab selected. A yellow box highlights a note: 'The list of mime-types explicitly included for processing by this source. If left empty, all records are indexed (default.)'. A red box highlights the radio button 'mime-types listed are excluded'. The 'Excluded MimeTypes' section is empty. To the right, there's a sidebar with buttons: 'addMimeType or File Extension' (highlighted in blue), 'add Video Types', 'add Audio Types', 'add Office Types', 'add Image Types', and 'Clear Excluded MimeTypes'. At the bottom are 'previous' and 'next' buttons, and at the very bottom are 'Close', 'Test', 'Reset Delta', and 'Save' buttons.

default settings, all files are processed in “exclusive” mode

In Exclusive mode (mime-types listed are excluded) you can add file-extensions or mimeTypes to the “Excluded MimeTypes” list. Any mimeType listed in this box is not processed, only “take account” off.

The list of mime-types explicitly excluded from being processed by this source. If left empty, all records go into the inventory.

only mimeType listed are included

Included MimeTypes *

Filter...

page 1 of 0 (0 items)

previous next

MimeTypes

mime-type / file-extension

add mimeType or File Extension

add Video Types

add Audio Types

add Office Types

add Image Types

Clear Included MimeTypes

Close Test Reset Delta Save

inclusive mode

Inclusive mode is the exact opposite. All files are now no longer processed but only taken in as part of SimSage’s inventory of what this source might contain. You still need to run / re-run the crawler for this particular source to get that information.

The list of mime-types explicitly included for processing by this source. If left empty, all records are indexed (default).

mimeType listed are excluded

Excluded MimeTypes

Filter...

T application/mp4

T video/mp4

page 1 of 1 (2 items)

previous next

MimeTypes

add mimeType or File Extension

add Video Types

add Audio Types

add Office Types

add Image Types

Clear Excluded MimeTypes

Close Test Reset Delta Save

example of processing all files, excluding MP4 types

The screenshot shows the 'MimeTypes' tab selected in a navigation bar. A note at the top states: 'The list of mime-types explicitly excluded from being processed by this source. If left empty, all records go into the inventory.' Below this is a toggle switch labeled 'only mime-types listed are included' which is checked. To its right is a text input field containing 'docx'. On the left, there's a list titled 'Included MimeTypes *' with a 'Filter...' input. The list contains one item: 'application/vnd.openxmlformats-officedocument.wordprocessingml.document'. Below the list are several buttons: 'addMimeType or File Extension' (highlighted with a red box), 'add Video Types', 'add Audio Types', 'add Office Types', 'add Image Types', and 'Clear Included MimeTypes'. At the bottom are pagination controls 'page 1 of 1 (1 items)', 'previous', and 'next'. At the very bottom are four buttons: 'Close', 'Test' (highlighted with a red box), 'Reset Delta', and 'Save'.

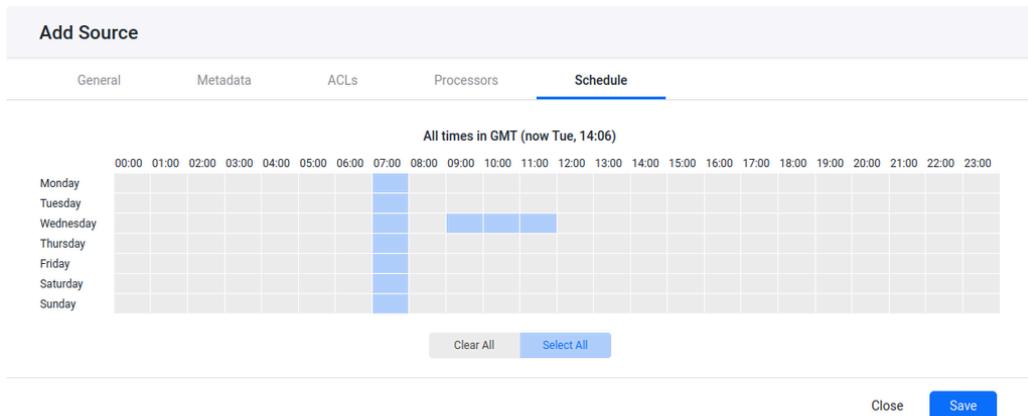
An example of only processing docx / Microsoft Office word documents

Button	Purpose
only mime-types listed are included / only mime-types listed are excluded	this toggle button switches between included (only mime-types in the list are processed) and excluded (default mode) (only mime-types in the list are not processed).
addMimeType or File Extension	this button, along with the text box after it, enables users to add file-extensions (e.g. docx, doc, mp4) or mimeType (e.g., application/mp4, text/plain) to the inclusion/exclusion list on the left.
add Video Types	this button adds all known video types to the list on the left. This includes many types such as mp4, mov, avi, etc.
add Audio Types	this button adds all known audio types to the list on the left. This includes many types such as mp3, wav, aiff, etc.
add Office Types	this button adds all known office types to the list on the left. This includes Microsoft Office, Open Office, Libre Office types, as well as Word, Excel, PowerPoint, and other equivalent known applications.
add Images Types	this button adds all known image types to the list on the left. This includes many types such as jpeg, png, gif, tiff, bmp etc.
Clear Included/Excluded MimeTypes	this button completely clears the list on the left

The Schedule Tab

Here you can specify when you would like the crawler to be allowed access to the SimSage platform.

This is done by setting desired time frames to active. In the example provided we used the “select all” button to allow SimSage to always be available to your crawler.



External Crawler Configuration

The external crawler is a small Java program, provided by SimSage for the version of your platform.

```
rock@rock-office:~/crawlers$ ll
total 24
-rwxrwxr-x 1 rock rock 1943 Jan 17 13:07 crawler.sh*
drwxrwxr-x 2 rock rock 12288 Jan 17 13:07 lib/
-rw-rw-r-- 1 rock rock 695 Jan 17 13:18 system.properties
rock@rock-office:~/crawlers$ 
```

The software provided consists of:

- A *lib* folder with all the java libraries required to run the crawler,
- A *crawler.sh* executable shell file for running the crawler,
- and a *system.properties* file that needs to be edited to match your platform.

The current version of the external crawler file is 7.6.3 and can be downloaded [here](#).

Once you have the external docker file, you will need to populate the values in the docker compose file as below. A docker-compose file might look like this:

```
1 version: "3.9"
2 services:
3   web:
4     build: .
5     deploy:
6       resources:
7         limits:
8           cpus: "1"
9           memory: 512M
10      reservations:
11        cpus: "1"
12        memory: 512M
```

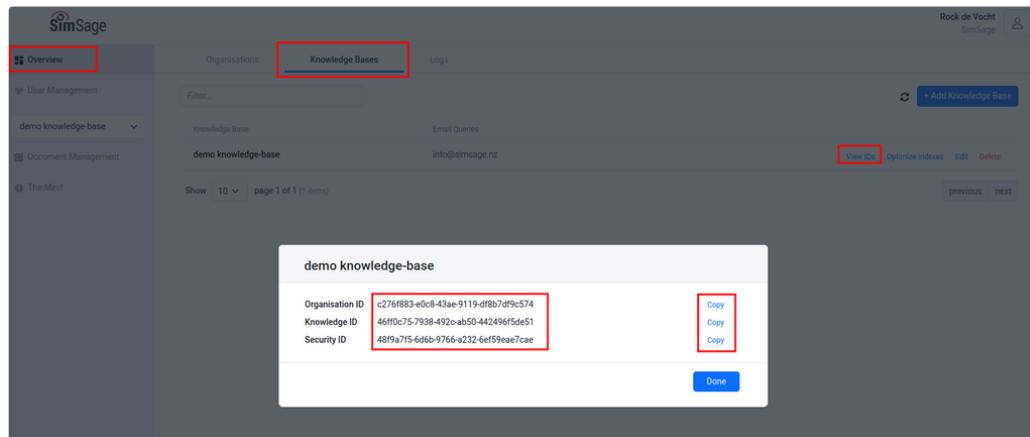
```

13   environment:
14     - source_id=1
15     - crawler_type=web
16     - organisation_id=c276f883-e0c8-43ae-9119-df8b7df9c574
17     - kb_id=46ff0c75-7938-492c-ab50-442496f5de51
18     - sid=48f9a7f5-6d6b-9766-a232-6ef59eae7cae
19     - simsage_endpoint=http://192.168.1.10:8080/api

```

The environment settings in bold must have the correct values supplied by SimSage and can be found via the SimSage Admin platform.

 The source_id should be set to the id of the internal crawler source you created in the above steps.

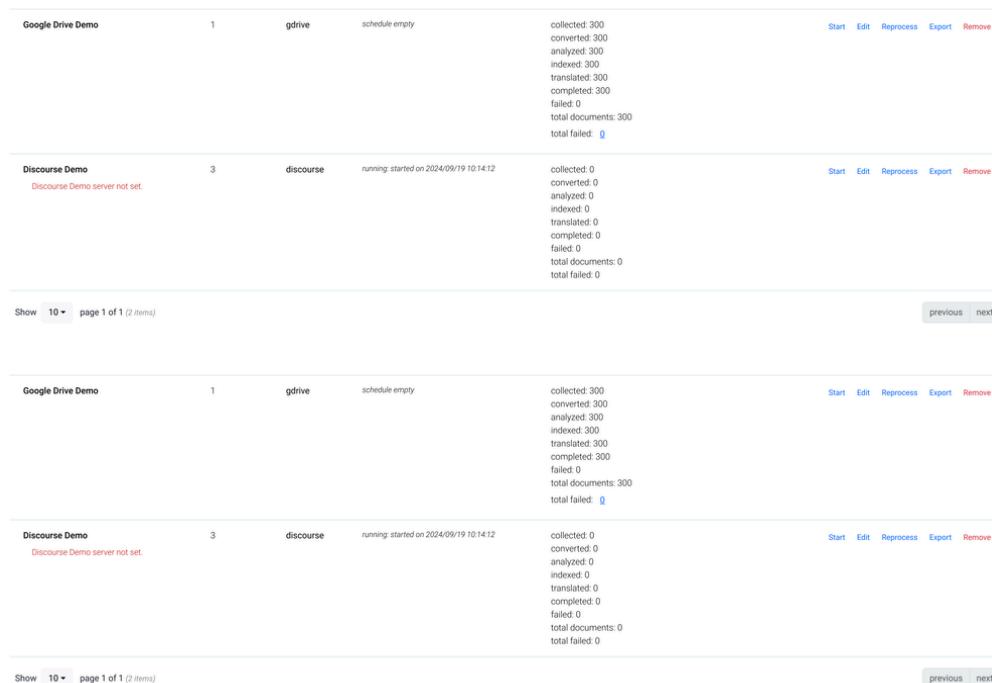


Troubleshooting

As part of setting up the crawler, there are three key checks users can perform to ensure everything is configured correctly:

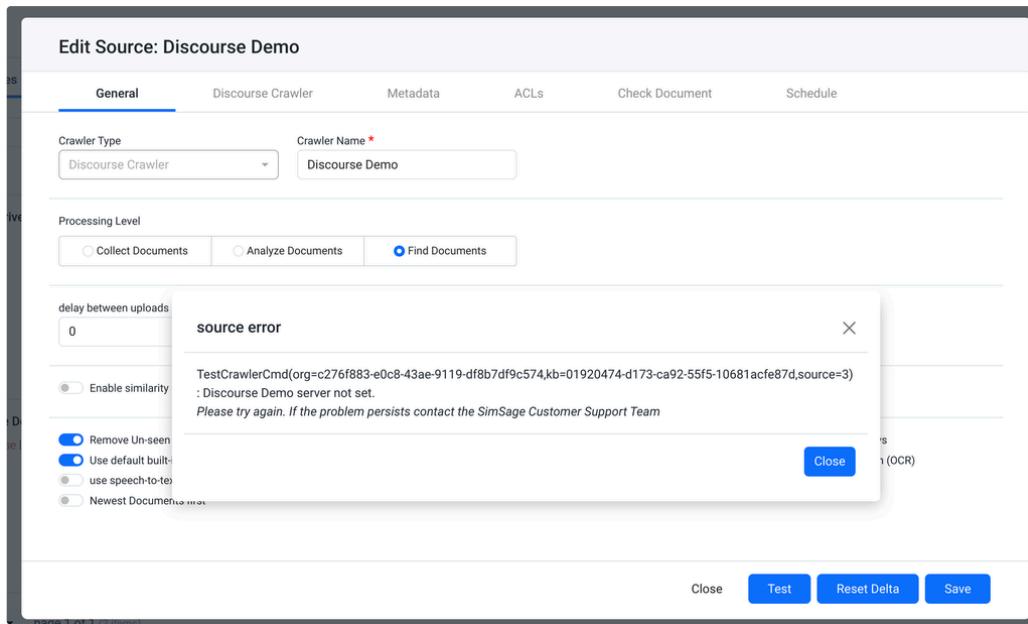
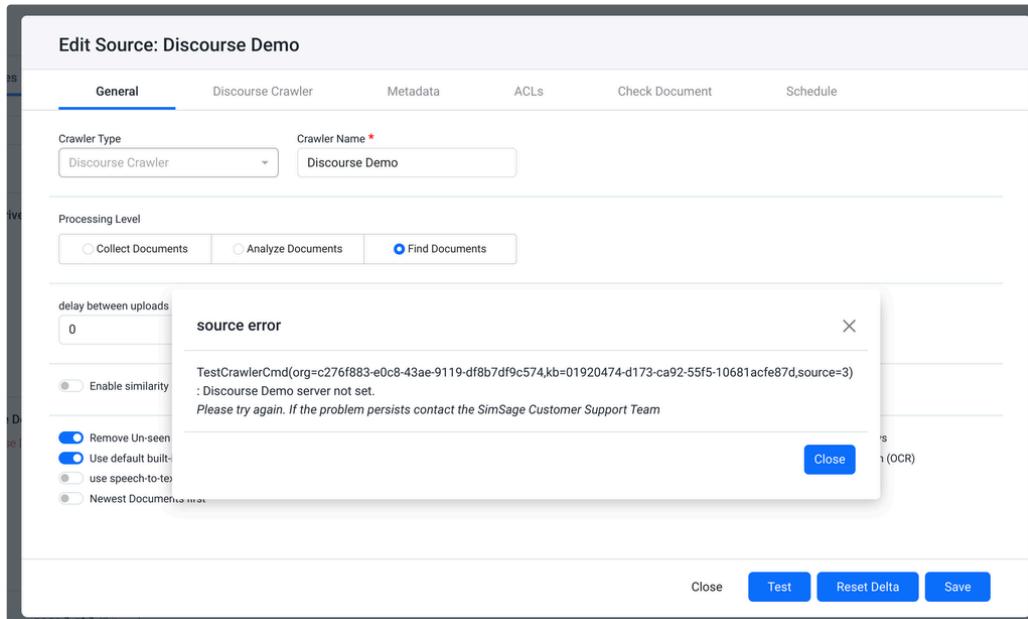
1. Error Messages in Crawler List

After saving the crawler setup, if SimSage detects any errors, they will be displayed beneath the crawler name in the list of crawlers. For example, a message might indicate that the Server details are missing.



2. Testing Platform Connectivity

If no errors are visible after saving, users can return to the crawler settings by selecting "Edit." Here, they will find a "Test" button, which allows them to verify if our platform can successfully communicate with the platform they are trying to connect to.



3. Reviewing Crawler Logs

By navigating to the "Overview" section from the navigation bar, users can access the "Logs" tab. From this section, they can review the crawler logs or any other service logs, and use the filter to search for specific keywords or log type that may help diagnose issues.

SimSage

Max Webster
SimSage

Overview User Management SimSage Demo Document Management The Mind

Organisations Knowledge Bases Logs Status

Select Service: Crawler Service Search: Filter:

Crawler log Show Entries Type Refresh

Downloaded 50 100 250 500 All Info Debug Error Warn Off 5 seconds 10 seconds

ERROR crawler 2024/09/19 10:21:51
n.a.c.s.comms.InternalCrawlerManager - fatal: runCrawler(c276f883-e0c8-43aa-9119-dfb7d9fc574.01920474-d173-ca92-55f5-10681acf87d.3 discourse:'Discourse Demo') java.security.InvalidParameterException: Discourse Demo server not set. at nz.simsage.crawlers.DiscourseCrawler.getParameters(DiscourseCrawler.kt:131) at nz.simsage.crawlers.DiscourseCrawler.test(DiscourseCrawler.kt:98) at nz.simsage.crawlers.DiscourseCrawler.run(DiscourseCrawler.kt:62) at nz.simsage.crawler.service.comms.InternalCrawlerManager.runCrawler(InternalCrawlerManager.kt:201) at nz.simsage.crawler.service.comms.InternalCrawlerManager.run\$lambda\$0(InternalCrawlerManager.kt:110) at kotlin.concurrent.ThreadsKt\$thread\$thread\$1.run(Thread.kt:30)

SimSage

Max Webster
SimSage

Overview User Management SimSage Demo Document Management The Mind

Organisations Knowledge Bases Logs Status

Select Service: Crawler Service Search: Filter:

Crawler log Show Entries Type Refresh

Downloaded 50 100 250 500 All Info Debug Error Warn Off 5 seconds 10 seconds

ERROR crawler 2024/09/19 10:21:51
n.a.c.s.comms.InternalCrawlerManager - fatal: runCrawler(c276f883-e0c8-43aa-9119-dfb7d9fc574.01920474-d173-ca92-55f5-10681acf87d.3 discourse:'Discourse Demo') java.security.InvalidParameterException: Discourse Demo server not set. at nz.simsage.crawlers.DiscourseCrawler.getParameters(DiscourseCrawler.kt:131) at nz.simsage.crawlers.DiscourseCrawler.test(DiscourseCrawler.kt:98) at nz.simsage.crawlers.DiscourseCrawler.run(DiscourseCrawler.kt:62) at nz.simsage.crawler.service.comms.InternalCrawlerManager.runCrawler(InternalCrawlerManager.kt:201) at nz.simsage.crawler.service.comms.InternalCrawlerManager.run\$lambda\$0(InternalCrawlerManager.kt:110) at kotlin.concurrent.ThreadsKt\$thread\$thread\$1.run(Thread.kt:30)