Web Archive

User Guide

August 28, 2024

Copyright Notice

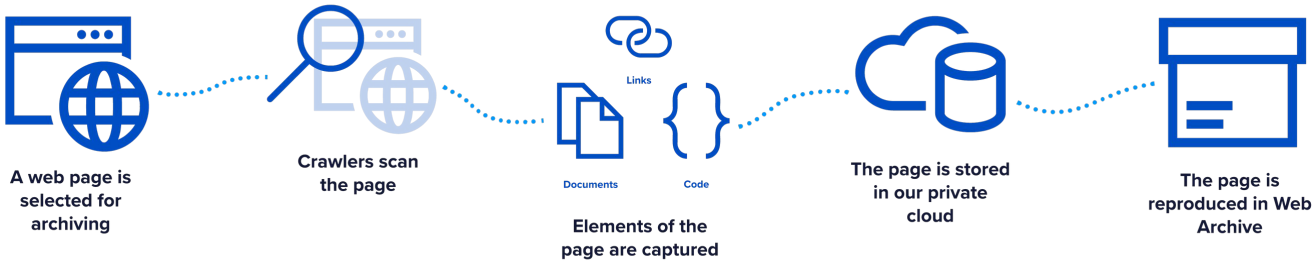
Document Disclaimer

Licensee is responsible for identifying the security options appropriate to the risks identified in licensee's environment and implementing those options.

Sample XML files and field values displayed in the documentation are examples only and should not be construed as appropriate for licensee's or end user's use.

The content in this PDF is subject to change without prior notice. Refer to docs.smarsh.com for the latest product documentation updates.

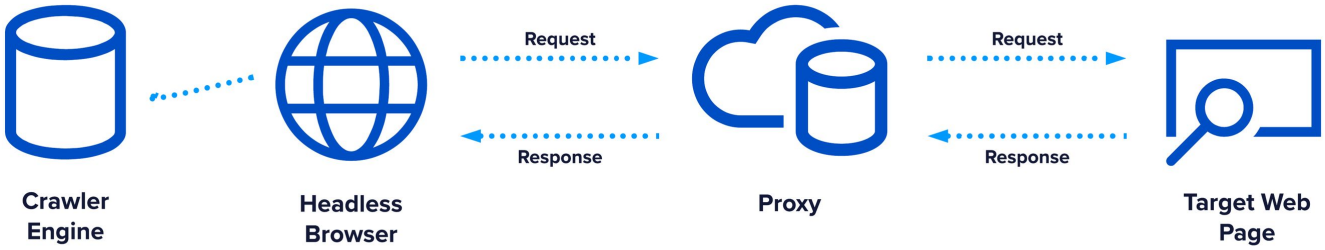
The following diagram provides a visual explanation of Web Archive.



Archiving architecture overview

Web Archive comprises several distinct components that perform its archiving functionality.

The following diagram provides a visual explanation of the archiving architecture powering Web Archive.



How archiving works

Implementation specialists first must configure timeout settings for the archiving process. This ensures archiving crawlers aren't running in perpetuity. The following timeout settings are demonstrative and not exhaustive:

- **Browser timeout**: The time it takes for the browser to load.
- **Load timeout**: The time for the browser to load the contents of the page.
- **Total archiving timeout**: The time it takes before the entire process times out.

Next, the Web Archive **crawler engine** opens a seed URL in a headless browser. Web Archive uses a **proxy** to grab the HAR file of the **target web page**.

The HAR file contains a page's HTML code and any accompanying resources needed to render the page. Captured resources include:

- CSS
- JavaScript
- Hyperlinks
- SRCs

Web Archive checks, validates, and processes all related resource links. It marks pages "archivable" if the **max page limit** isn't hit and there aren't exclusionary **URL filters**. URL filters tell the system to exclude or include specific web pages.

The archiving process continues until the max page limit is finally hit.

Best practices for creating archivable websites

There is no singular, comprehensive set of design specifications to ensure full compatibility with capture and replay services such as Web Archive. You must instead ensure your archived websites are as easy to capture and replay as possible. The following best practices will help you design the most-compatible websites.

Authentication

Web Archive supports a simple username and password authentication process. The crawler can't impersonate different users during its processing. If you need to capture content behind authentication, you must ensure all the content requiring capture is available to the user credentials provided to Smarsh. There is currently no way to capture personalized, per-user content.

The following are general authentication best practices:

- Make sure username and password fields use an HTML `id` or `name` attribute on the authentication page.
- Make the authentication page an explicit HTML page. Don't use pop-ups.
- Make sure the authentication page sends the HTML username and password information via HTML form submission.
- Ensure that the form submission action is an HTML button that's uniquely identified by an HTML `id` or `name` attribute.
- Don't use CAPTCHA or any secondary authentication methods, including two-factor.
- Don't use the browser runtime instance to fingerprint the authentication token
- Set a minimum authentication token expiration time of 12 hours.
- Multiple paid tiers of users with unique tiered content require multiple configured crawls; one per user tier.

Cookies

Cookies help Web Archive communicate states to your website. Cookies are the recommended method to suppress pop-up windows, headers, footers, or other modal-type interactions.

Note

Web Archive doesn't support setting or altering cookies between crawling processes or during active crawling. You must configure your cookies prior to Web Archive crawling your pages.

The following are general cookie best practices:

- Make cookie values static: don't use dates or input a random value.
- Make cookie values simple and easy to validate.
- Don't use cookies as passwords.
- Cookies should keep the same name and value pairing across your site.

CRM/CMS platforms

Follow these best practices for interacting with Customer Relationship Management (CRM) or Content Management System (CMS) platforms:

- Limit the retrieval of data and content from CRM/CMS sites.
    - CRM/CMS sites use custom code and AJAX/XHR requests for data retrieval, which can prevent proper archiving.
- Sites like SharePoint, WIX, and Squarespace are mostly dynamic platforms and are incompatible with Web Archive. SharePoint has alternative capture methods, but Wix, Squarespace, and others do not.
- Ensure your CMS platform always writes the URLs with a permalink.

Documents

Documents include files and other assets with their own file extensions and URLs.

- Refer to documents directly with their file extension. Avoid using query strings in the URL for the document. Properly identifying the URLs reduces the time to identify the content and therefore reduces the chance of timeouts.
- Prevent the retrieval of documents through AJAX/XHR calls. The content may load at various times during the archive process, making the capturing of documents using these calls not guaranteed.

Dynamic content

Dynamic content changes based on the individual user and is generated upon request.

- URLs dynamically constructed during page load may not be captured.
- AJAX/XHR requests for additional data during page load may not be captured. They also will not be replayed when recreating the webpage.
- Use HTML rather than JSON content as HTML content is what is parsed and altered during replay.
- Ensure that w3 want archived is preloaded and not dependent upon dynamic user interaction. This is the best option for capturing post-load dynamic content.

Dynamic Link Discovery

Dynamic Link Discovery (DLD) is the inspection of a web page to find URLs that are hidden or visible. Using DLD with a seed URL doesn't guarantee the discovery and processing of required pages.

The following are general DLD sitemap best practices:

- Make sure URL `<href>` or `<src>` attributes are discoverable in the DOM after page load.
- Web Archive might not capture URLs dynamically created during page load.
- AJAX/XHR requests for additional data during page load may not be captured. That means they wouldn't be available for replay.

Elements

Modals, pop-up windows, headers, and footers are web page elements that users often interact with. They can be dynamic and static.

- Use cookies to suppress these elements when possible.
- Modals in general can cause page timeouts through failed "on-load" events. Limit your use of modals.
- The SmarshBot user agent can be used for modal or pop-up window suppression. However, depending upon your CDN or web server configuration, it may suppress or block traffic.

Indexes

The index page is a URL or file that automatically loads when a web browser starts.

- Make sure you include the content you wish to search against inside the source HTML of the page.
- Make sure the source HTML includes a title tag minimally.

Local Storage

Local storage allows Web Archive to communicate states to your JavaScript page. You can also use local storage to suppress pop-ups windows, headers, footers, or other modals.

> **Note**
>
> Web Archive doesn't support setting or altering local storage between crawling processes or during active crawling. You must configure your local storage settings prior to Web Archive crawling your pages.

The following are general local storage best practices:

- Make local storage values static: don't use dates or input a random value.
- Make local storage values static: don't use dates or input a random value.
- Make local storage values simple and easy to validate.
- Storage entries should keep the same name and value pairing across your site.

Multimedia content

Multimedia content includes videos and other animated web files.

- Use HTML5-compatible content instead of a custom media player.
- Officially supported video platforms include:
    - Vimeo
    - Wistia
    - Brightcove
    - YouTube (YouTube sometimes blocks Smarsh IP addresses and can result in inconsistent capture).
- Use the <iframe> method of embedding video content if native capture or replay is required. JavaScript players obscure the URL and can't be captured.
- Video can be captured as <href> attributes for anchor tags.
- The most reliable way to replay video is to create a simple anchor tag for the video and allow the browser to recognize the file as video and replay it in its own environment.
- Note that if you still prefer to use a JavaScript player, Web Archive cannot duplicate that user experience during replay. However, any captured video is available in an export.

Post-page load processing

- Plugins can cause content to render after the browser has given the `on_load` signal. This can prevent capture.

Screenshot support

Web Archive takes screenshots as part of its archival process. You can export pages as screenshots using the UI. Consider the following recommendations to ensure the best screenshots.

- Change your pages' background image CSS properties from `vh` (variable height) values to `px` (pixel size) values to prevent unexpected cropping.
- Avoid scrolling headers and footers when possible. There are limitations on screenshot heights and sizes. As a result, for pages that exceed those heights, multiple images are taken and stitched together. Floating headers and footers will appear multiple times when images are stitched together and may obstruct page content.
- Ensure the browser window reports the proper maximum height. Screenshots may be smaller than desired if incorrect height values are returned.

Seed URLs

The Web Archive crawling process always begins with the provided seed URL. Web Archive supports two kinds of seeds:

- XML-based sitemaps
- Dynamic Link Discovery

> **Note**
>
> Sitemaps are the recommended seed URL output. They are deterministic and not dynamically created which allows for greater archival accuracy.

Throttling

The following best practices can help prevent the prevention or slowdown of Web Archive as it captures your sites:

- Some CDNs and other proxies identify Smarsh traffic as coming from a bot. You can work around this behavior by adding the Smarsh IP address space to an "allowed" list.
- Throttling can also occur automatically when your servers are overwhelmed with bot requests. If you're using a sitemap, properly size your hardware to ensure that all the listed pages can be requested simultaneously.
- The smallest unit of parallelism that Smarsh can guarantee is 20 simultaneous page requests.

Timeouts

Timeouts occur when Web Archive is unable to continue its capturing processes. Avoid timeouts by ensuring that the server is sized correctly to handle parallel requests.

The following best practices can prevent this from occurring:

- The default timeout for capturing pages is 30 seconds. However, low-performing servers can often become slow and unresponsive.
- Avoid having your JavaScript code aggregate large quantities of raw data, which causes significant download times.
- Do not suppress the "on-load" event

URLs

Managing the format and syntax of your URLs ensures proper capture of web pages.

- Keep URLs as absolute as possible to ensure that they can be found and replaced during replay.
- Avoid adding GUIDs or timestamps as query parameters. This causes all pages to be marked as changed. It also removes the ability for Web Archive to replace these URLs during replay.
- Avoid partial URLs when possible. While convenient in your JavaScript, it makes replay more difficult to contextualize and archive.
- Avoid dynamically created URL generation and injection into the DOM. Dynamic URLs break replay.
- Ensure that downloadable resources maintain their file extension. A URL without a file extension may be misidentified as a page and result in a timeout.

User interaction to access content

- User interactions will not take place during archiving.
- If content requires a user interaction to show specific page content, then there is a risk that it may not be captured during archiving.

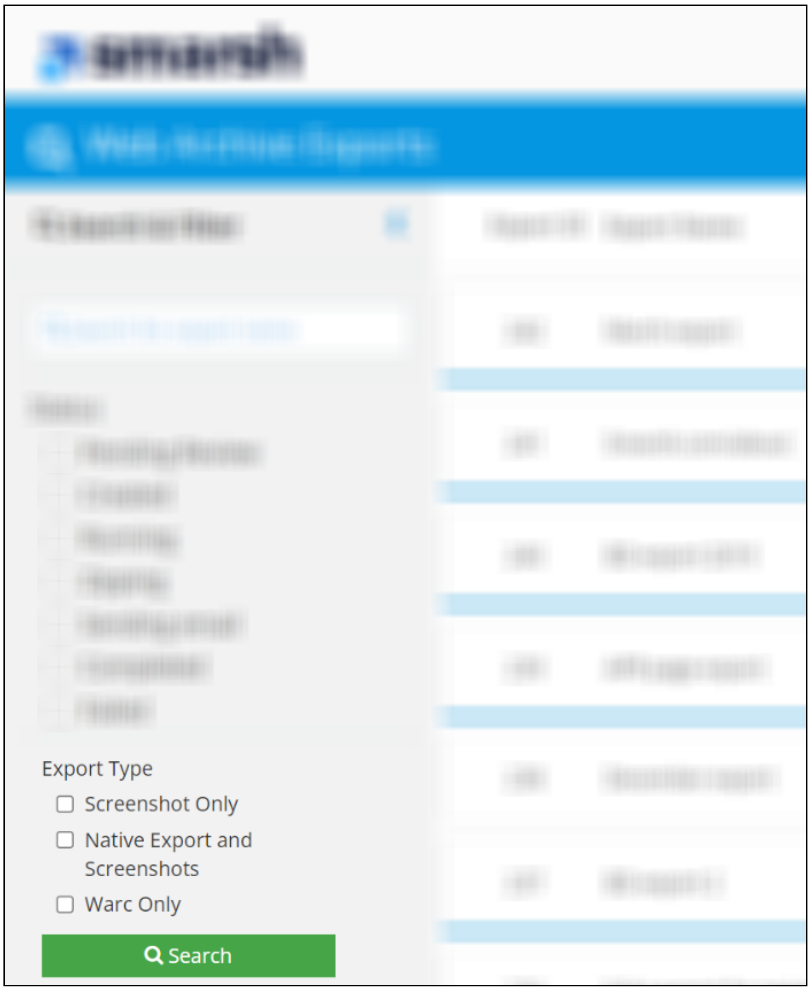
XML sitemaps

Sitemaps are always the preferred type of seed URL because you can identify a starting set of pages for archiving.

> **Note**
>
> Sitemaps only specify the starting set of web pages you want archived. It doesn't stop Dynamic Link Discovery from occurring later in the process on the pages specified in your sitemap.

Web Archive sitemap processing code was written to the specification found on sitemaps.org. Complying with this sitemap specification ensures a greater likelihood of compatibility and support while using Web Archive.

Export Type filters



Like with the **Status** filters, you can find exports by their **Export Type**. Click one or all check boxes and then click the **Search** button. An export name isn't needed to search using **Export Type** filters.

Full-data export packages

A full-data export includes the underlying code and all the screenshots of web pages in a crawl. See Self-service exports for more information on creating your own exports. Complete exports may also include images and other file assets recovered during the export process.

Note

Custom exports created by the Client Data team are zipped and encrypted. You'll need to unzip the files and provide a password before you can access the export contents. Smarsh support emails the password required to unencrypt your export.

Note

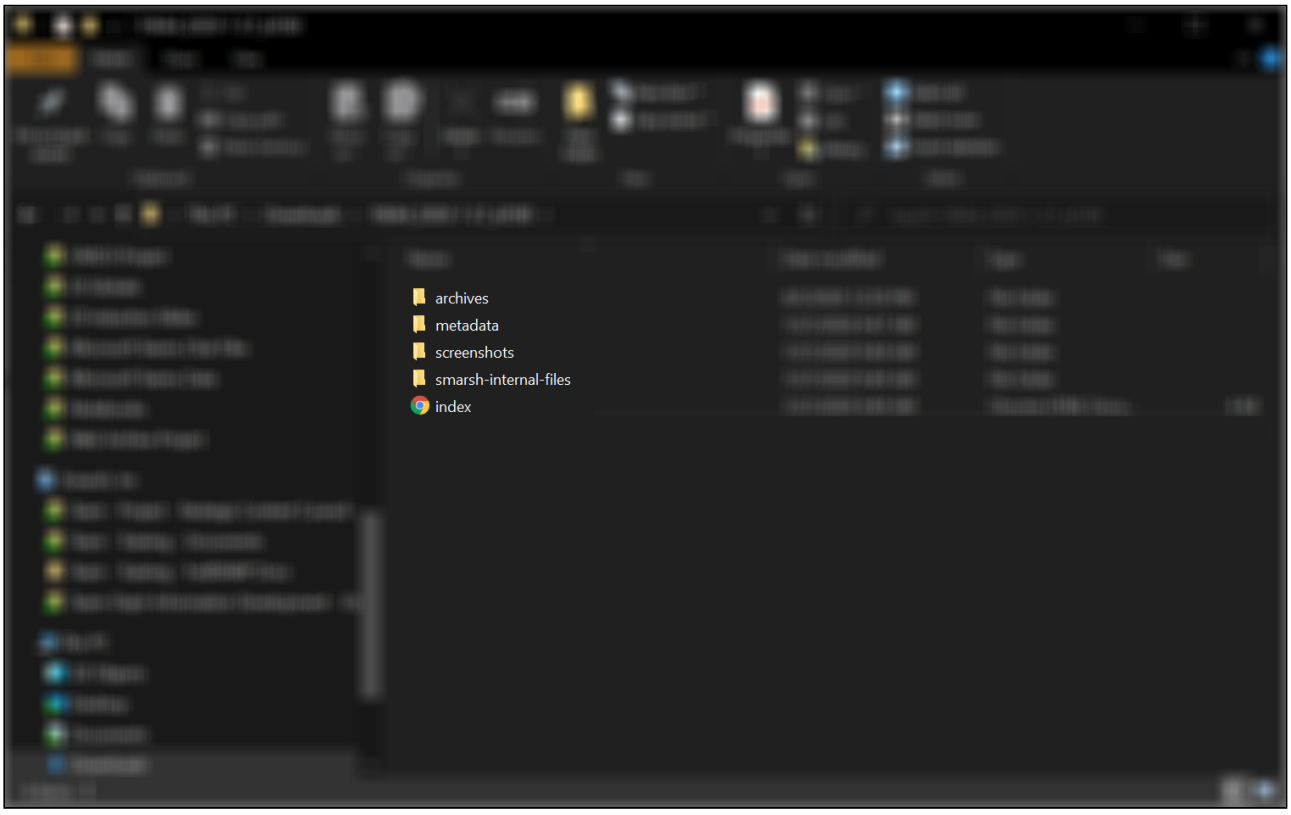
Screenshot-only exports are another option available to self-service exports or through the Client Data team. These export packages don't contain the underlying code resources of a full-data export.

The full-data export package folder structure

Full-data exports contain several folders comprising the data of your export. Exports also include an HTML-based navigation tool usable in any web browser.

The structure of a full-data export folder is the same whether you created your own export, or received a physical media export in the mail from Smarsh support.

Package contents



All export packages have the same folder structure. After unzipping and unencrypting the export, you'll see the following folders and HTML file:

- `archives`
- `metadata`
- `screenshots`
- `smarsh-internal-files`
- `index.html`

Table 1: Export package folder definitions

Title	Definition
archives	Each individual file comprising the selected crawl in the export package.

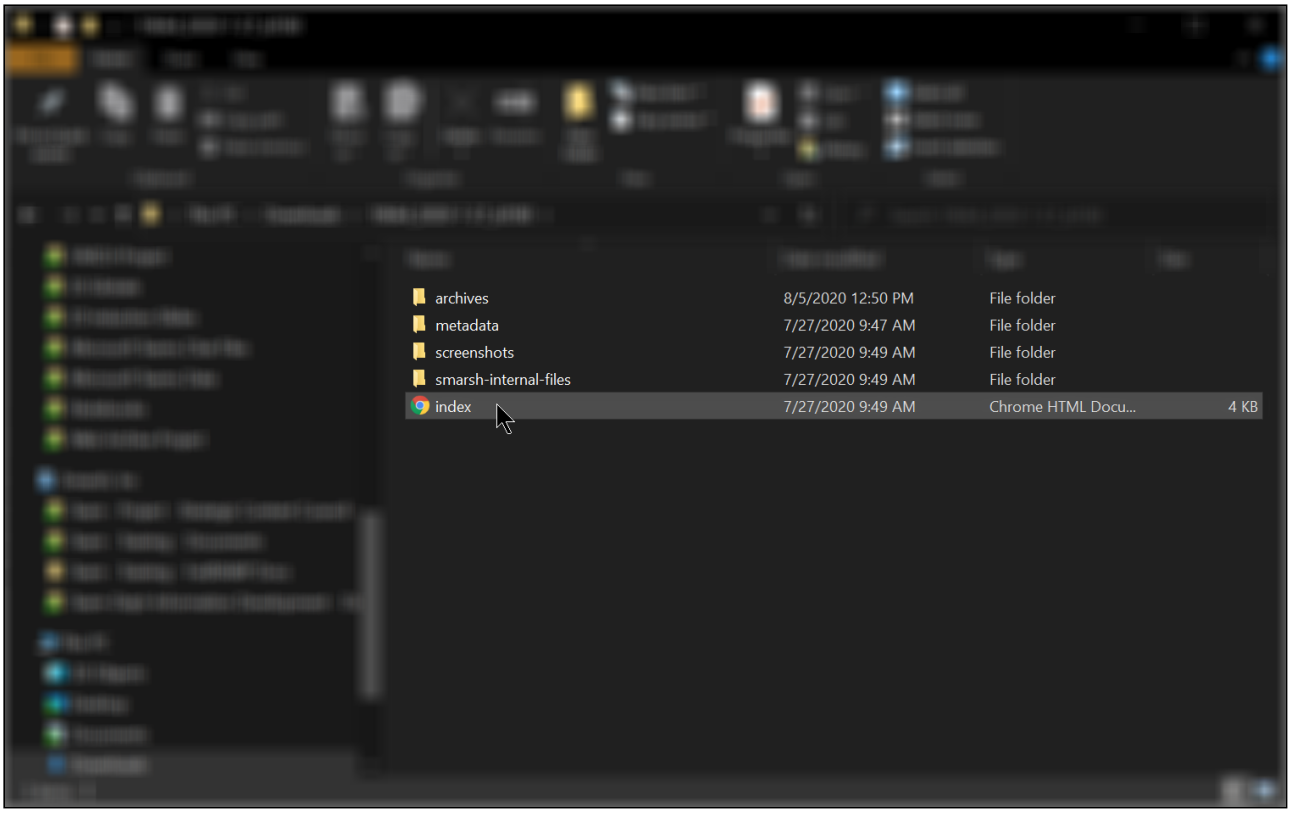
Title	Definition
metadata	Contains a **resources** subfolder. Inside that folder are HTML summary files for each web page captured in the crawl. You can open these files in your web browser to view the underlying resources captured in the export.
screenshots	Every screenshot Web Archive captured during the exported date range. These screenshots are image files without file extensions. You can use the built-in export browser to view them. See Browsing the full-data export package for more information.
smarsh-internal-files	Assets supporting the built-in export browser.
index.html	Opens the built-in export browser in your default web browser. See Browsing the full-data export package for more information.

Note

You can create a screenshot-only export or request one through the Client Data team. If you requested a screenshot-only export, your package will look different. An index.html file is still provided for browsing you export.

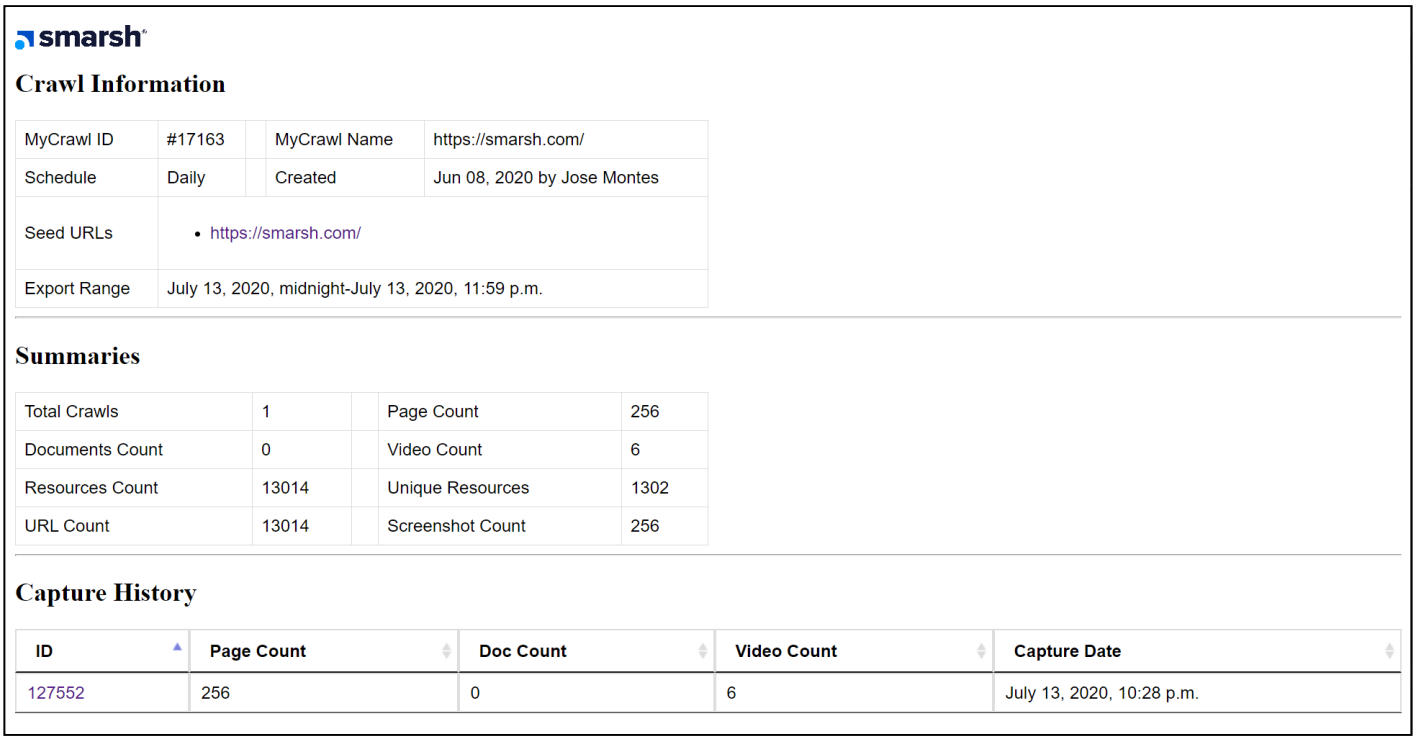
Browsing the full-data export package

Each export package includes an index.html file. Open this file in any web browser to review the contents of your export.



The export browser

Use the export browser to interact with your full-data export via a web page.



smarsh

Crawl Information

MyCrawl ID	#17163	MyCrawl Name	https://smarsh.com/
Schedule	Daily	Created	Jun 08, 2020 by Jose Montes
Seed URLs	• https://smarsh.com/		
Export Range	July 13, 2020, midnight-July 13, 2020, 11:59 p.m.		

Summaries

Total Crawls	1	Page Count	256
Documents Count	0	Video Count	6
Resources Count	13014	Unique Resources	1302
URL Count	13014	Screenshot Count	256

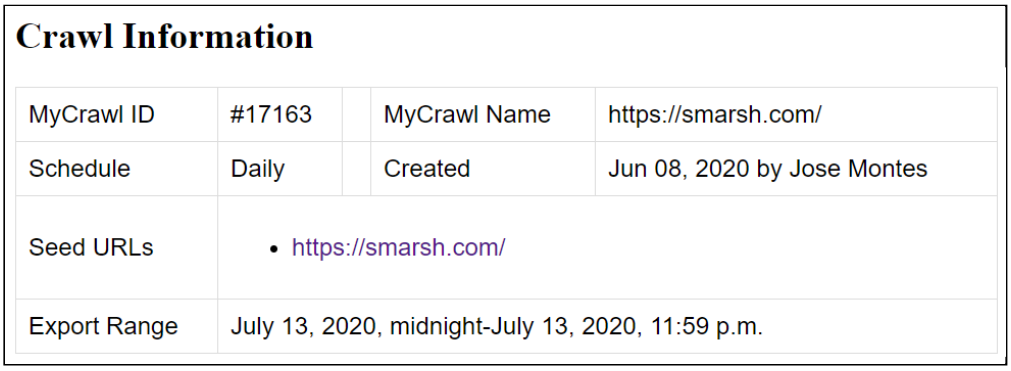
Capture History

ID	Page Count	Doc Count	Video Count	Capture Date
127552	256	0	6	July 13, 2020, 10:28 p.m.

The export browser includes the following sections:

- Crawl Information
- Summaries
- Capture History

Crawl Information



Crawl Information

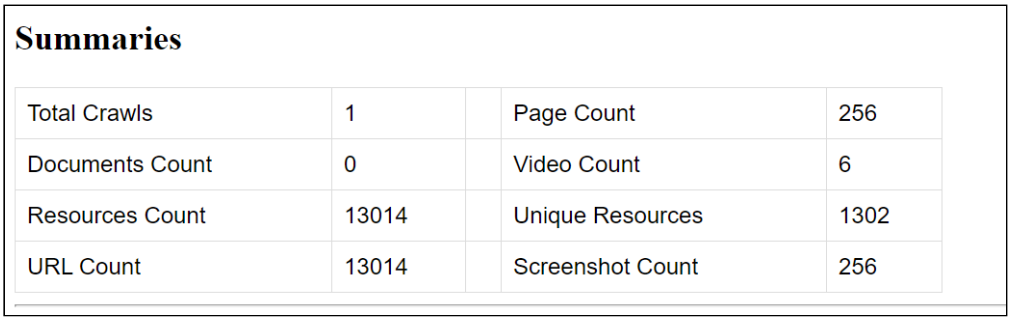
MyCrawl ID	#17163	MyCrawl Name	https://smarsh.com/
Schedule	Daily	Created	Jun 08, 2020 by Jose Montes
Seed URLs	• https://smarsh.com/		
Export Range	July 13, 2020, midnight-July 13, 2020, 11:59 p.m.		

This table contains identifying information about the exported, singular crawl including your account number, the schedule of the crawl, and the date range of the export.

Table 1: The Crawl Information table definitions

Title	Definition
MyCrawl ID	The unique ID of the crawl in the export. A MyCrawl is a set of instructions defining the type of capture, frequency of capture, the URLs captured, and more.
MyCrawl Name	The name of the crawl given at configuration.
Schedule	The frequency of the crawl. Crawls run daily, weekly, or at any other cadence set during implementation.
Created	The creation date of the export.
Seed URLs	Which URLs constitute the exported crawl. The crawl in the image above only archives one URL: smarsh.com.
Export Range	The full date and time range of the crawl contained in the export.

Summaries



Summaries

Total Crawls	1		Page Count	256
Documents Count	0		Video Count	6
Resources Count	13014		Unique Resources	1302
URL Count	13014		Screenshot Count	256

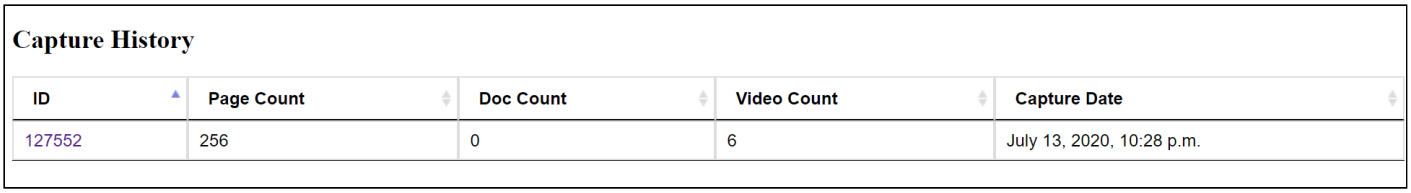
This table provides an overview of page changes during the exported date range.

Table 2: The Summaries table definitions

Title	Definition
Total Crawls	The number of archive crawls in the export.
Documents Count	The number of documents captured in the crawler date range. Documents can include Word files, PDFs, and others.
Resources Count	The number of items, including images and code files, that make up a particular page.
URL Count	The total unique URLs scanned that branch from the provided seed URL.

Title	Definition
Page Count	The number of individual web pages scanned during the crawl.
Video Count	The number of videos archived during the crawl. Videos include .mp4, .wmv, and others.
Unique Resources	The unique count of resources that make up a particular page. Some pages reuse images or code files; this counts only unique resources.
Screenshot Count	The number of screenshot captures completed during the crawl. Screenshot files are included in the export package.

Capture History


Capture History

ID	Page Count	Doc Count	Video Count	Capture Date
127552	256	0	6	July 13, 2020, 10:28 p.m.

This table provides details on the specific capture, including the date it completed and the number of pages, documents, or videos captured.

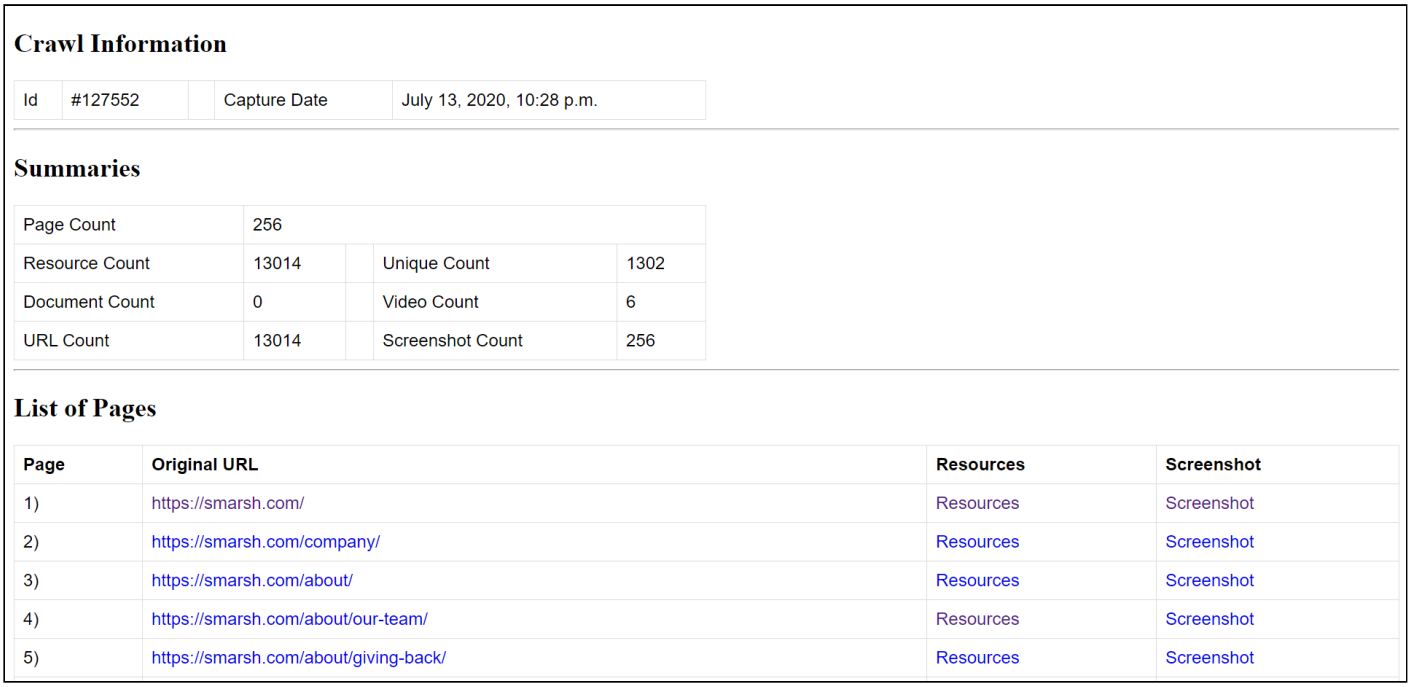
Table 3: Capture History table definitions

Title	Definition
ID	The unique ID for a specific crawl.
Page Count	The number of individual web pages scanned during the crawl.
Doc Count	The number of documents captured in the crawler date range. Documents can include Word files, PDFs, and others.
Video Count	The number of videos archived during the crawl. Videos include .mp4, .wmv, and others.
Capture Date	The date the specific crawl completed its scan.

Clicking the unique ID of the crawl takes you to the crawl summary browser for that specific crawl.

The crawl summary browser

The crawl summary page summarizes a single crawl. It includes links to captured pages, underlying code resources, and any associated screenshots.



Crawl Information

Id	#127552	Capture Date	July 13, 2020, 10:28 p.m.

Summaries

Page Count	256		
Resource Count	13014	Unique Count	1302
Document Count	0	Video Count	6
URL Count	13014	Screenshot Count	256

List of Pages

Page	Original URL	Resources	Screenshot
1)	https://smarsh.com/	Resources	Screenshot
2)	https://smarsh.com/company/	Resources	Screenshot
3)	https://smarsh.com/about/	Resources	Screenshot
4)	https://smarsh.com/about/our-team/	Resources	Screenshot
5)	https://smarsh.com/about/giving-back/	Resources	Screenshot

The crawl summary browser includes the following sections:

- Crawl Information
- Summaries
- List of Pages

Crawl Information

Crawl Information

Id	#127552	Capture Date	July 13, 2020, 10:28 p.m.

This table provides summary information about the single crawl.

Table 1: The Crawl Information table definitions

Title	Definition
Id	The unique ID for a specific crawl.
Capture Date	The date the specific crawl completed its scan.

Summaries

Summaries

Page Count	256			
Resource Count	13014		Unique Count	1302
Document Count	0		Video Count	6
URL Count	13014		Screenshot Count	256

This table provides numerical counts of captured assets from the crawl.

Table 2: The Summaries table definitions

Title	Definition
Page Count	The number of individual web pages scanned during the crawl.
Resource Count	The number of items, including images and code files, that make up a particular page.
Document Count	The number of documents captured during the crawl. Documents can include Word files, PDFs, and others.
URL Count	The total unique URLs scanned that branch from the provided seed URL.
Unique Count	The unique count of resources that make up a particular page. Some pages reuse images or code files; this counts only unique resources.
Video Count	The number of videos archived during the crawl. Videos include .mp4, .wmv, and others.
Screenshot Count	The number of screenshot captures completed during the crawl. Screenshot files are included in the export package.

List of Pages

List of Pages

Page	Original URL	Resources	Screenshot
1)	https://smarsh.com/	Resources	Screenshot
2)	https://smarsh.com/company/	Resources	Screenshot
3)	https://smarsh.com/about/	Resources	Screenshot
4)	https://smarsh.com/about/our-team/	Resources	Screenshot
5)	https://smarsh.com/about/giving-back/	Resources	Screenshot

This table lists all the pages captured during the crawl.

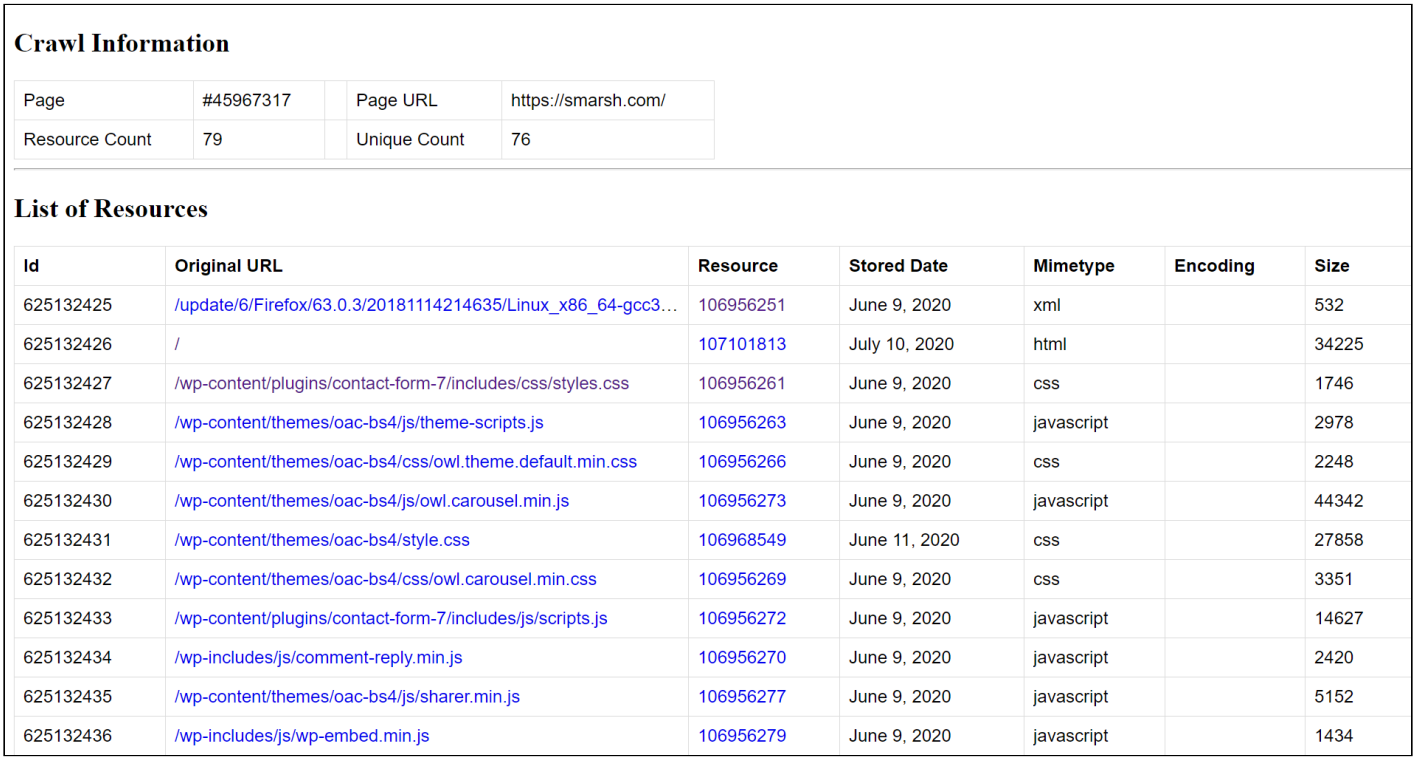
Table 3: The List of Pages table definitions

Title	Definition
Page	A descending numbered list of the pages captured in the crawl.
Original URL	A relative URL. This URL differs from the crawl summary browser's `List of Pages` attribute.
Resources	A direct link to the the resources browser, which lists code resources like CSS and JavaScript files.
Screenshot	A direct link to the captured screenshot of the page.

Clicking the **Resources** link opens the resources browser. Clicking the **Screenshot** link opens the captured image in a new browser tab.

The resources browser

The resources browser details the underlying page assets captured during the crawl. Crawls capture CSS, XML, JavaScript, and other common web assets.



Crawl Information

Page	#45967317	Page URL	https://smarsh.com/
Resource Count	79	Unique Count	76

List of Resources

Id	Original URL	Resource	Stored Date	Mimetype	Encoding	Size
625132425	/update/6/Firefox/63.0.3/20181114214635/Linux_x86_64-gcc3…	106956251	June 9, 2020	xml		532
625132426	/	107101813	July 10, 2020	html		34225
625132427	/wp-content/plugins/contact-form-7/includes/css/styles.css	106956261	June 9, 2020	css		1746
625132428	/wp-content/themes/oac-bs4/js/theme-scripts.js	106956263	June 9, 2020	javascript		2978
625132429	/wp-content/themes/oac-bs4/css/owl.theme.default.min.css	106956266	June 9, 2020	css		2248
625132430	/wp-content/themes/oac-bs4/js/owl.carousel.min.js	106956273	June 9, 2020	javascript		44342
625132431	/wp-content/themes/oac-bs4/style.css	106968549	June 11, 2020	css		27858
625132432	/wp-content/themes/oac-bs4/css/owl.carousel.min.css	106956269	June 9, 2020	css		3351
625132433	/wp-content/plugins/contact-form-7/includes/js/scripts.js	106956272	June 9, 2020	javascript		14627
625132434	/wp-includes/js/comment-reply.min.js	106956270	June 9, 2020	javascript		2420
625132435	/wp-content/themes/oac-bs4/js/sharer.min.js	106956277	June 9, 2020	javascript		5152
625132436	/wp-includes/js/wp-embed.min.js	106956279	June 9, 2020	javascript		1434

The resources browser contains the following sections:

- Crawl Information
- List of Resources

Crawl Information

Crawl Information

Page	#45967317	Page URL	https://smarsh.com/
Resource Count	79	Unique Count	76

This table summarizes the captured resources associated with a particular page.

Table 1: The Crawl Information table definitions

Title	Definition
Page	This is the unique ID of the page you clicked in the crawl summary browser to reach the resource browser.
Resource Count	The total number of resources associated with the page.
Page URL	The URL of the page whose assets you're viewing.
Unique Count	The unique count of resources that make up a particular page. Some pages reuse images or code files; this counts only unique resources.

List of Resources

List of Resources

Id	Original URL	Resource	Stored Date	Mimetype	Encoding	Size
625132425	/update/6/Firefox/63.0.3/20181114214635/Linux_x86_64-gcc3…	106956251	June 9, 2020	xml		532
625132426	/	107101813	July 10, 2020	html		34225
625132427	/wp-content/plugins/contact-form-7/includes/css/styles.css	106956261	June 9, 2020	css		1746
625132428	/wp-content/themes/oac-bs4/js/theme-scripts.js	106956263	June 9, 2020	javascript		2978
625132429	/wp-content/themes/oac-bs4/css/owl.theme.default.min.css	106956266	June 9, 2020	css		2248
625132430	/wp-content/themes/oac-bs4/js/owl.carousel.min.js	106956273	June 9, 2020	javascript		44342
625132431	/wp-content/themes/oac-bs4/style.css	106968549	June 11, 2020	css		27858
625132432	/wp-content/themes/oac-bs4/css/owl.carousel.min.css	106956269	June 9, 2020	css		3351
625132433	/wp-content/plugins/contact-form-7/includes/js/scripts.js	106956272	June 9, 2020	javascript		14627
625132434	/wp-includes/js/comment-reply.min.js	106956270	June 9, 2020	javascript		2420
625132435	/wp-content/themes/oac-bs4/js/sharer.min.js	106956277	June 9, 2020	javascript		5152
625132436	/wp-includes/js/wp-embed.min.js	106956279	June 9, 2020	javascript		1434

This table identifies specific resources, including their file type and size.

Table 2: The List of Resources table definitions

Title	Definition
Id	A unique ID for each listed resource.

Title	Definition
Original URL	A relative URL. This URL differs from the crawl summary browser's `List of Pages` attribute.
Resource	A direct link to the listed code. Clicking this link opens the code file in a new tab of your browser.
Stored Date	The original date that version of the file was downloaded and stored.
Mimetype	The media type of the resource. Valid mimetypes include `xml`, `html`, `css`, `javascript`, and others.
Encoding	The encoding type of the resource, if applicable.
Size	The size of the resource in kilobytes.

Single-page exports

You can choose to export one or multiple pages using the **Action Panel**.

- Exporting a single page
- Exporting multiple pages

Note

Only the **Screenshot**, **Text**, and **Screenshot & Text** options allow you to export multiple pages simultaneously. The **Screenshot to PDF** option only exports the page currently selected in the **Action Panel**, regardless of how many boxes are checked.

Exporting a single page

To export a page:

1. Run a new search.
2. Select an archived page to open the **Action Panel**.
3. Click the **More** drop-down button in the top-right corner of the **Action Panel**.