Web Scraper Chrome Extension. Case Studies.

Tamizdat Project: Banned Books Now. Basic Scraper

https://tamizdatproject.org/publications/banned-books-now/

1. Website Analysis

Before creating selectors, analyze the target website:

Navigate to the target page and explore the structure

Identify pagination/loading mechanisms:

- Does the content load on scroll?

- Are there "Load More" buttons?

- Is there traditional pagination?

For Tamizdat: It's not the case, as the publication page loads automatically.

Data hierarchy: Publications section → Individual book pages, so no need to add any extra selectors to navigate to other pages except for the book's page.

2. Create New Sitemap

Open the target URL in your browser

→ Right-click anywhere on the page

→ Select "Inspect" to open Developer Tools

→ Navigate to the "Web Scraper" tab

→ Click "Create new sitemap"

→ Enter sitemap details:

→ Name: tamizdat_banned_books

→ Start URL: https://tamizdatproject.org/publications/banned-books-now/

Save sitemap

3. Book Link

Collect links to individual book pages

Add new selector

→ Selector ID: book_links

→ Type: Link

→ Parent: _root

→ Multiple: Yes (for full dataset), select 2 or more books / No (for testing), select 1

→ Save selector

This selector targets individual book elements

4. Book Metadata Selectors

Once on individual book pages, create selectors for each data point:

Open the first book page

Click on the book_links selector

Scraping book title:

Add new selector

→ Selector ID: book_title

→ Type: Text

→ Parent: book_links

→ Multiple: No

→ Save selector

This selector targets title element

Scraping publisher information:

Add new selector

→ Selector ID: publisher

→ Type: Text

→ Parent: book_links

→ Multiple: No

→ Save selector

This selector targets the publisher element

Scraping book cover image:

Add new selector

→ Selector ID: book_cover

→ Type: Image

→ Parent: book_links

→ Multiple: No

→ Save selector

This selector targets the cover image element

5. Testing and Scraping

5.1 Initial Test Scrape

Run the scraper with a limited scope (single book)

Review results in the data browser

Verify data quality

5.2 Full Dataset Scraping

Once testing is successful, edit the book_links selector

→ Check "Multiple" to collect all books

→ Save changes

→ Run a full scrape

6. Data Review

Click "Browse" to examine scraped data

Click "Export data"

Select format

Download the file

Kultura. Advanced Scraping: Customized selectors

https://kulturaparyska.com/pl/publication/2/year/1947

1. Website Analysis

Before creating selectors, analyze the website:

Year-based pagination: Each year has its own dedicated page

Hover interactions: Journal content appears when hovering over publications

Rich metadata: Each publication includes covers, descriptions, and content tables

The website uses predictable URL patterns for different years, which allows targeted scraping.

For a single year:

start url: https://kulturaparyska.com/pl/publication/2/year/1947

But if we want only the 1965-1970 issues:

start url: https://kulturaparyska.com/pl/publication/2/year/[1965-1970]

2. Create New Sitemap

Open the target URL in your browser

→ Right-click anywhere on the page

→ Select "Inspect" to open Developer Tools

→ Navigate to the "Web Scraper" tab

→ Click "Create new sitemap"

→ Name: kultura_[start_year]_[end_year]

Example: kultura_1965_1970

→ Start URL: https://kulturaparyska.com/pl/publication/2/year/[1965-1970]

→ Save sitemap

3. Journals Collection

3.1. Wrapper Selector

→ Id: wrapper_selector

→ Type: Element (scroll)

→ Selector: #publication-type-2 div.col-3

→ Multiple: Yes

* Set to "Multiple" because there are many publications per page

→ Parent Selectors: _root

Selector breakdown:

Creates the structural foundation for collecting multiple publications

#publication-type-2: Targets the main publications container

div.col-3: Selects individual publication cards in the grid

3.2 Content Metadata Selectors

Tap on wrapper_selector

Table of Contents Selector

→ ID: get_table

→ Type: Text

→ Selector: div.contents

→ Multiple: No

→ Parent: wrapper_selector

Selector breakdown:

Extracts publication descriptions and table of contents that appear on hover

div.contents: targets div elements with the "contents" class, works within each publication card to get specific metadata

Cover Image Selector

→ ID: get_cover

→ Type: Image

→ Selector: img

→ Multiple: No

→ Parent: wrapper_selector

Selector breakdown:

Captures publication cover images for each journal issue

img: Targets all image elements

4. Testing and Scraping

Use "Data preview" to verify results

Scrape and export the dataset