Linked Art Tool Documentation

Overview

This script is designed to interface with data compliant with Linked Art. It fetches Linked Art JSON data from a provided URL and processes it to extract and display key metadata in a reader-friendly format. The script runs in a command-line interface (CLI) environment.

Prerequisites

- Node.js and npm must be installed on your machine.
- Required npm packages: node-fetch, fs, and js-yaml.

Installation *(skip if prerequisites have been fulfilled)*

Install Node.js and npm:

- Download and install Node.js from nodejs.org.
- npm (Node Package Manager) is included with Node.js.

Install required packages:

- Run the following command to install the required packages:
  npm install node-fetch fs js-yaml

Usage

- Navigate to the directory in which the tool is installed.
- The script is executed in the command line using the command:
  node latool.js <URL> [OPTIONS]
- ***<URL>*** is the URL of the Linked Art JSON data you want to process

Options

- ***--log***: Include this argument to display log messages for any issues that occurred during processing. Among other issues, this may include broken links, non-standard structuring practices, and non-standard controlled vocabulary term usage.
- ***--concise***: Include this argument to output only one form of the work type, timespan, dimensions, and materials information.
- ***--found***: Include this argument to skip output of fields where no data is found.
- ***--save=<filename>***: Include this argument to save the output to a YAML file in the same directory as the tool. Replace <filename> with your desired file name followed by '.yaml' (e.g. ***--save=output.yaml***).
- NB: options can be combined, e.g. to save a concise version of a Linked Art URL with the log included:
  node latool.js http://example.com/data.json --concise --log --save=output.yaml

Output

Currently, the tool supports the following fields, all of which are displayed by default:

1. Title
2. Exhibited title
3. Former title
4. Accession number
5. Creator(s)
6. Work type (classification)
7. Work type (statement)
8. Timespan (name)
9. Timespan (structured)
10. Dimensions statement
11. Dimensions (structured)
12. Materials statement
13. Materials (structured)
14. Location
15. Owner
16. Set
17. Social media (I modelled this for the *Labyrinth* objects but it's not a standard field)
18. Credit line
19. Citations
20. Access statement
21. Description
22. Provenance description
23. Web page(s)
24. IIIF manifest
25. Primary image
26. Primary thumbnail
27. All images
28. All thumbnails

- To omit a field from the display, the *outputOrder* object can be modified in the script, similar to the *object_display_order* object in objects.yaml in the Quire extension.
- When the ***--concise*** argument is used, fields 6, 8, 10, and 12 are prioritised over fields 7, 9, 11, and 13, which are treated as their alternate or secondary forms; this can easily be swapped around in the code if desired however.