Data Dictionary (FOLIO Schema Parser)

Overview

We have a Google Spreadsheet that stores links to FOLIO app schemas and examples and parses them to create a consolidated list of modules, interfaces, and data elements.

To use:

- Click on the "Review Data Elements" tab
- Scroll and search to find data elements of interest
- If desired, use a temporary filter to filter down to particular interfaces or modules

Preview

(Note: for all interface features, make sure to go to the full Google Spreadsheet.)

Technical Details

The spreadsheet is owned by Angela Zoss, and a few other people have permissions to make edits and refresh the data.

The spreadsheet uses Google Apps Script for Google Sheets to retrieve schema and example data from FOLIO GitHub repositories.

Steps used to populate data:

- In the "Add Interfaces" tab, paste in the module name, interface path, LDP table name, URL to schema JSON, URL to example JSON (optional), and URL to API documentation for every LDP table.
- Either using a manual refresh trigger or the automatic overnight trigger*, the Apps Script executes a parse routine.
  - Open the "Add Interfaces" tab and copy all of the data
  - Clear the current "Review Data Elements" tab
  - Loop through each row from the "Add Interfaces" tab
    - Fetch the examples URL and store the resulting JSON object in a global dictionary containing all of the examples, organized by interface name.
    - Store the module, interface, and LDP table names from the first tab to use in every row.
    - Create a header row for each interface, using just the module, interface, and LDP table name.
    - Fetch the JSON object from the schema URL and extract the data associated with the "properties" key. (This should be an object with a series of key/value pairs where each key is a data element and each value is another object that contains "type" and "description" for almost all data elements.)
    - For each key under "properties", build a row for that key that includes the type and description from its value object.
      - If the property has a type of "object," need to look for properties of that object and loop through them as well.
      - If the property has a type of "array," need to check if it's an array of objects, in which case loop through those as well.
    - Combine all rows into a big array of arrays, then use that to set the values of the "Review Data Elements" tab, starting with row 2.
    - Add formatting (alternating gray rows, green background, links, notes, and bold font for table header rows).

*Note: This script, because it "fetches" data from one or more URLs, can only be run after authenticating as the owner of the spreadsheet. It can't just run using automatic processes like onOpen(). So, an editor can refresh it manually while the spreadsheet is open, or the spreadsheet can be set to refresh on a "trigger" - something that tells Google to run the script after a certain condition is true or at specified time intervals. The script is currently set to be triggered overnight.

There is also a backup of the Google Apps Script shared on GitHub.