

The individual scripts

Jonathan Hooper, Leeds. May 2022

Contents

Overview	1
Alternative	1
Running the scripts	1
(1) Collect reading list citations.....	1
(2a) Enhance citations with data from Alma	2
(2b) Enhance citations with data from Scopus, WoS, VIAF	2
(3) Process data and export spreadsheets.....	2
simpleExport.php writes:	3
longExport.php writes:.....	3
Note on the Summary.CSV file.....	3

Overview

The process uses a collection of standalone PHP scripts that each carry out individual steps.

There are essentially three stages:

1. Collect reading list citations from Leganto
2. Enhance citations with data from Alma, Scopus, WoS, VIAF
3. Process data and export spreadsheets

What the scripts have in common:

- They all (except for the first step) expect an input of JSON-encoded citation data on STDIN
- They all (except for the final step) produce output of JSON-encoded citation data on STDOUT
- They all (except the first and final steps) work by adding data from an external source to the input citations, and outputting the enhanced citations
- They share utility functions in **utils.php** and configuration in **config.ini** and **Config/**

Alternative

*Instead of running the individual scripts, it may be simpler to use the **batch script** to process multiple modules and steps using a single command. See **1_4_the_batch_script.pdf**.*

Running the scripts

(1) Collect reading list citations

This collects relevant metadata (authors, titles, DOIs, Alma MMS IDs etc) from reading list citations in Alma/Leganto.

Run `getCitationsByModule.php` specifying the module(s) as a parameter, and receiving the output from STDOUT e.g.:

```
php getCitationsByModule.php -m PSYC3505 >Data/tmp/PSYC3505_L.json
```

This uses the Ex Libris Courses API to find courses with a searchable ID matching PSYC3505, finds all matching reading lists, and writes JSON-encoded citation data from these lists to `Data/tmp/PSYC3505_L.json`

(2a) Enhance citations with data from Alma

This enhances the Leganto citations with bibliographic metadata from any linked records in Alma (authors, titles, ISBN, ISSN etc)

Run the script `enhanceCitationsFromAlma.php` providing citation data on STDIN and receiving the output from STDOUT e.g.:

```
php enhanceCitationsFromAlma.php <Data/tmp/PSYC3505_L.json >Data/tmp/PSYC3505_A.json
```

This decodes the JSON-encoded citations in the input file, and loops over all the citations. Where a citation contains an Alma MMS ID it retrieves metadata from Alma using the Ex Libris Bibliographic Records API, and adds this to the citation. Finally it writes the JSON-encoded enhanced citations to the output file.

(2b) Enhance citations with data from Scopus, WoS, VIAF

This enhances the citations with metadata from the external sources Scopus, WoS and VIAF. It is principally focussed on collecting geographical data on authors, and on calculating the similarity between the author/title in the reading list with that in the external source.

See "Searching the data sources" for more information on how these external sources are searched.

Run the scripts `enhanceCitationsFromScopus.php`, `enhanceCitationsFromWos.php`, `enhanceCitationsFromViaf.php` providing citation data on STDIN and receiving the output from STDOUT e.g.:

```
php enhanceCitationsFromScopus.php <Data/tmp/PSYC3505_A.json >Data/tmp/PSYC3505_S.json
php enhanceCitationsFromWos.php <Data/tmp/PSYC3505_S.json >Data/tmp/PSYC3505_W.json
php enhanceCitationsFromViaf.php <Data/tmp/PSYC3505_W.json >Data/tmp/PSYC3505_V.json
```

For each integration, this decodes the JSON-encoded citations in the input file, and loops over all the citations. Where appropriate, the external source (Scopus/WoS/VIAF) is searched and retrieved data added to the citation. Finally it writes the JSON-encoded enhanced citations to the output file.

The order may be significant - currently, the VIAF integration may make use of data from the Scopus and WoS integrations and so it is best run at the end.

(3) Process data and export spreadsheets

This takes the enhanced citations with Leganto, Alma, Scopus, WoS and VIAF data, and produces spreadsheet reports to send to the Library and Project staff and for them to send to tutors.

Run the two scripts `simpleExport.php` and `longExport.php`, providing citation data to each on STDIN. Unlike the other scripts, these two scripts do not write to STDOUT, but instead to CSV files in **Data/** e.g.:

```
php simpleExport.php <Data/tmp/PSYC3505_V.json
php longExport.php <Data/tmp/PSYC3505_V.json
```

These decode the JSON-encoded citations in the input file, and after some processing on them they each write data to CSV files, suitable for viewing in Excel by non-technical staff. The files are saved to the directory **Data/**

`simpleExport.php` writes:

- For each reading list, a spreadsheet **Data/{LIST-CODE}.CSV** with a row-per-citation of author-country data from the reading list and the external source
 - This sheet optionally also includes a separate table below the main one giving the total counts for each country in that list
- A single spreadsheet, **Data/Summary.CSV** with a row-per-reading list giving a list-level summary

`longExport.php` writes:

- For each reading list, a longer report **Data/{LIST-CODE}_LONG.CSV** with multiple rows-per-citation, giving all the geographical data obtained for each citation, a separate piece of information on each row

Note on the `Summary.CSV` file

Using the commands above, the `Summary.CSV` file is rewritten each time, and will only contain rows for that particular module. It is more useful to produce a single `Summary.CSV` file with entries for multiple modules, so the `simpleExport.php` script supports `-i` (initialise) and `-a` (append) options:

To produce a single `Summary.CSV` file listing reading lists for two modules, PSYC3505 and BLGY3135, run the following:

```
php simpleExport.php -i
php simpleExport.php -a <Data/temp/PSYC3505_V.json
php simpleExport.php -a <Data/temp/BLGY3135_V.json
```

The initial **`simpleExport.php -i`** command does not write any reports, but creates an empty `Summary.CSV` file with the column headings.

The two **`simpleExport.php -a`** commands create the reports as normal, but the per-reading-list summary is **appended** to the existing `Summary.CSV` rather than creating a new one.

The batch script (see [1_4_the_batch_script.pdf](#)) takes care of this initialising and appending so that a single `Summary.CSV` is automatically generated, covering all the modules in the batch.

For more information on the export process see [3_1_export_scripts.pdf](#).