# IMLS Search and Compare

# Dataset Import Documentation

## Overview:

The documentation below outlines the process for importing three types of data into the Algolia search service for the IMLS Search and compare tool (<https://www.imls.gov/labs/>)

* Library systems data
* Outlet data
* Longitudinal/trends data from the current and past surveys

The scripts are written in the Ruby programming library. Ruby comes pre-installed on Mac computers and can be installed on Windows computers (see Prerequisites section).

Once the scripts are run and JSON files are created, the files need to be uploaded to the Algolia search engine for indexing and eventual consumption by the Search and Compare application.

The scripts are available in the IMLS github repository in the data folder. The IMLS github repository for the Search and Compare project is located at:

<https://github.com/IMLS/library-search-compare>

## Prerequisites

**Ruby**: Ruby is a free, open-source programming language that runs on Windows, MacOS and Linux operating systems. Ruby can be downloaded and installed at: <https://www.ruby-lang.org/en/downloads/>

It is also possible to run multiple version of Ruby on a single computer using the Ruby Version Manager (RVM). Instructions for installing RVM are located at: <https://rvm.io/rvm/install>

The scripts below were run on version ruby-2.4.1, but should also run on newer versions of Ruby.

**Ruby Gems:** Ruby functionality can be extended via library packages called Gems. The scripts below use the following Gems:

* json
* csv
* charlock\_holmes

To install Ruby Gems type the following at a command or terminal prompt: gem install [gem\_name]

Ruby will install the gem and any dependencies.

To display a list of installed gems type: gem list

**Search and Compare Ruby scripts:** Ruby scripts have a .rb extension. The scripts used for the Search and Compare application are contained in the data\_scripts folder. To execute a ruby script, open the command line or terminal application and type:

ruby [script\_name].rb

**Algolia:** Algolia is the service the Search and Compare uses to create a search index. It is where the json data files are uploaded after being processed by the Ruby scripts. Currently there are no IMLS staff members who have access to the Search and Compare account on Algolia. Mark can add IMLS staff to the Algolia “Team”.

**GitHub**: The code for both the Ruby scripts and the Search and Compare application are contained in the IMLS library-search-compare GitHub repository located at: <https://github.com/IMLS/library-search-compare>

IMLS is the owner of the GitHub repository and has the ability to add and remove team members. However, you can also contact Mark to manage team members.

The data folder in the screenshot below contains the Ruby scripts (be sure that the master Branch is selected in the dropdown above).

A screenshot of a cell phone

Description automatically generated

## Folder structure

The following folder structure is used for the dataset processing scripts which correspond to the type of data being processed.

* **current\_year** – Contains data used for the main search page and the individual library details page.
* **longitudinal** – Contains data that populates the Trends section of the library details page that enables comparison to data from years past.
* **outlets** – Contains data that populates the Outlets (branches, bookmobiles, etc.) section of the library details page.

# Current Year data:

## Download IMLS Public Library Survey (PLS) Data

**Procedure:**

* Go to the IMLS Public Library Survey page:
  + <https://www.imls.gov/research-evaluation/data-collection/public-libraries-survey>
* Scroll down to the appropriate FY
* Click on the CSV link and save the zip file
* Unzip the file
* Copy the Administrative Entity (AE) file to the current\_year folder
* Copy the outlet data file to the outlets folder

## Convert the Current Year PLs Data to JSON

**Script Name:** current\_year\_csv\_to\_json.rb

**Procedure:**

* Verify that the csv file was downloaded to the current\_year folder.
* In a text editor, open the current\_year\_csv\_to\_json.rb file.
* Change the value of @csv\_file in line #5 to the filename of the downloaded csv file  
  
* Open a command line application to the current\_year folder if it isn't currently open to that location.
* Execute the ruby script: ruby current\_year\_csv\_to\_json.rb

**Result:**

* Two json files are created:
  + current\_year.json
  + current\_year\_pretty.json
* The two files contain the same data. The \_pretty version is formatted to be human readable.
* You can ignore the \_working.csv file

**Notes:**

* The formatted FY17 json file is 28MB, the unformatted file is 22.6MB.
* The formatting script does the following:
  + Convert the fields listed in the fields\_to\_f array on line 10 from strings to numbers
  + Remove records that have hours listed as -3
  + Convert values of -1 to M
  + Convert values of -9 to S
  + Convert locale codes to labels i.e. 11, 12 and 13 become City

## Import Library Systems Data into Algolia

**File to be uploaded:** current\_year.json

**Procedure:**

* Login to the Algolia Indices management page:
  + <https://www.algolia.com/apps/CDUMM9WVUG/explorer/indices>
* The index used for library systems data is: libraries
* Click on the libraries link in the list of indices
* On the Manage index drop down, choose "Clear"
  + Warning, this will remove all of the records in the index
  + Until the new data is uploaded and indexed there won’t be any searchable records
* Once the current records are cleared, open the Add records drop down and choose "Upload file"
* Upload the current\_year.json file

**Result:**

* The current\_year.json file will be uploaded and indexed
* There were 9,216 records uploaded for the FY17 library systems data

**Note:**

* Below is a screenshot of Algolia management page. The links to the three main indices are at the top (outlets, libraries, trends).

A screenshot of a cell phone

Description automatically generated

# Outlets Data:

## Download imls Library Outlets Data

**Procedure:**

* Go to the IMLS Public Library Survey page:
  + <https://www.imls.gov/research-evaluation/data-collection/public-libraries-survey>
* Scroll down to the appropriate FY
* Click on the CSV link and save the zip file
* Unzip the file
* Copy the outlet data file to the outlets folder

## Convert the Outlets Data to JSON

**Script Name:** outlet\_csv\_to\_json.rb

**Procedure:**

* Verify that the outlets csv file was copied to the outlets folder.
* In a text editor, open the outlet\_csv\_to\_json.rb file.
* Change the value of @csv\_file in line #5 to the filename of the downloaded csv file  
  
* Open a command line application to the outlets folder if it isn't currently open to that location.
* Execute the ruby script: ruby outlet\_csv\_to\_json.rb

**Result:**

* Two json files are created:
  + outlets.json
  + outlets\_pretty.json
* The two files contain the same data. The \_pretty version is formatted to be human readable.
* You can ignore the \_working.csv file

**Notes:**

* The formatted FY17 json file is 16.6 MB, the unformatted file is 12.8 MB.
* The formatting script does the following:
  + Convert the fields listed in the fields\_to\_i array on line 5 from strings to numbers
  + Converts hours and weeks open listed -3 to 0
  + Convert values of -1 to M

## Import Outlets Data into Algolia

**File to be uploaded:** outlets.json

**Procedure:**

* Login to the Algolia Indices management page:
  + <https://www.algolia.com/apps/CDUMM9WVUG/explorer/indices>
* The index used for outlets data is: outlets
* Click on the outlets link in the list of indices
* On the Manage index drop down, choose "Clear"
  + Warning, this will remove all of the records in the index
  + Until new data is uploaded and indexed, the Search and Compare application won’t display any outlets data
* Once the current records are cleared, open the Add records drop down and choose "Upload file"
* Upload the outlets.json file.

**Result:**

* The outlets.json file will be uploaded and indexed
* There were 17,452 records uploaded for the FY17 outlets data

# Longitudunal/Trends Data:

## Download imls Library Data csv files

**Procedure:**

* Go to the IMLS Public Library Survey page:
  + <https://www.imls.gov/research-evaluation/data-collection/public-libraries-survey>
* Scroll down to the current FY
* Click on the CSV link and save the zip file
* Unzip the file
* Copy the Administrative Entity (AE) file to the longitudinal folder
* Repeat the steps above for the previous year, 5 years back and 10 years back

## convert the csv Data files and combine them into one Json file

**Script Name:** longitudinal\_csv\_to\_json.rb

**Procedure:**

* Verify that the current year, previous year, 5 and 10 year csv files were copied to the longitudinal folder.
* In a text editor, open the longitudinal\_csv\_to\_json.rb file.
* Change the value of @current\_year in line #6 to the current year of the data being uploaded  
  
* Change the values of the csv files in lines #7 - #10 to the filenames of the AE csv data files  
  A black sign with white text

  Description automatically generated  
  The correct filename will need to be added for each entry (i.e. in 2019, the names of the 2019, 2018, 2014 and 2009 csv files will need to replace the values for @current\_csv, @one\_year\_csv, @five\_year\_csv and @ten\_year\_csv respectively)
* Open a command line application to the longitudinal folder if it isn't currently open to that location.
* Execute the ruby script: ruby longitudinal\_csv\_to\_json.rb

**Result:**

* Two json files are created:
  + trends.json
  + trends\_pretty.json
* The two files contain the same data. The \_pretty version is formatted to be human readable.
* You can ignore the \_working.csv files

**Notes:**

* The longitudinal\_csv\_to\_json.rb file takes a few minutes to run. It will output some text to the screen for each FY data file it is processing.
* The formatted FY17 trends json file is 22.5 MB, the unformatted file is 18 MB.
* The formatting script does the following:
  + Remove records that have hours listed as -3
  + Convert values of -1 to M
  + Convert values of -9 to S

## Import Trends Data into Algolia

**File to be uploaded:** trends.json

**Procedure:**

* Login to the Algolia Indices management page:
  + <https://www.algolia.com/apps/CDUMM9WVUG/explorer/indices>
* The index used for library outlets data is: longitudinal
* Click on the longitudinal Link in the list of indices
* On the Manage index drop down, choose "Clear"
  + Warning, this will remove all of the records in the index
  + Until the new data is uploaded the search and compare application won’t display any Trends data.
* Once the current records are cleared, open the Add records drop down and choose "Upload file"
* Upload the trends.json data file

**Result:**

* The trends data file will be uploaded and indexed
* For FY17 there were 9,216 records uploaded to the longitudinal index

## Testing the data changes

Once that json data files have been uploaded to Algolia, you can test if the data has been updated by going to the Search and Compare site, clicking on a library and verifying that new data is being displayed on the details page. You may need to reload the browser page to see the new data. The \_pretty.json files can help to verify if the new data is being shown. Alternatively, each Algolia index page provides a nice display of the data – and it is easily searchable to find the library data you are verifying.

## Code Changes to switch the Fiscal YEar on the site

To change the fiscal year displayed in the Search and Compare application:

* Open the js/common.js file
* Change value for the current\_year variable in line #2 to the correct year:  
  