# inBloom Published Implementer Toolkit

## inBloom Published Overview

The inBloom Published program is developed for content providers and maintainers who wish to ensure their digital content is discoverable in the next-generation of tools for personalized education. inBloom Published ensures content providers use new metadata standards and search technologies correctly end-to-end including tagging content to metadata (LRMI) and learning standards (CCSS), publish to the Learning Registry (LR) and are discoverable through inBloom’s Learning Registry Index (LRI). The goal of the program is to increase both the quantity and relevancy of content discoverable in the LRI.

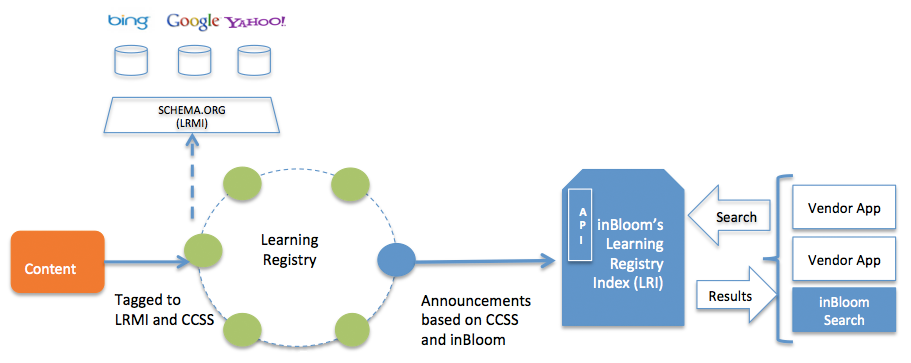
inBloom Published benefits for publishers:

* Expand market opportunities with use of the inBloom Published logo and brand.
* Increase access to inBloom customer base.
* Increase discoverability and use of content and products, including:
  + Learning maps tools
  + Content recommendation engines
  + Lesson planning applications
  + Content collaboration tools
* Co-marketing via inBloom website, collateral and marketing events.

This document outlines a publisher toolkit including guidance to publish an existing metadata database into the Learning Registry and inBloom’s Learning Registry Index. This describes a low-effort method of publishing into the Learning Registry by using the steps below:

* Export from an existing metadata catalog/database or out of a Microsoft Excel spreadsheet into a Common Separated Value (CSV) spreadsheet. This spreadsheet also provides a section for standards alignment.
* Use the provided script to convert the metadata from CSV to LRMI files, a format necessary for the Learning Registry and Learning Registry Index.
* Use the provided script to submit the LRMI files in bulk to the Learning Registry.
* Validate that the metadata has been submitted to the Learning Registry and Learning Registry Index.

## Digital Content Technology Overview



## Publishing to the Learning Registry

The Learning Registry has two environments available:

Sandbox (for testing purposes): <http://sandbox.learningregistry.org/>

Production: <http://node01.public.learningregistry.net/>

It is recommended that you first experiment and verify results are as expected before publishing into a production node of the Learning Registry.

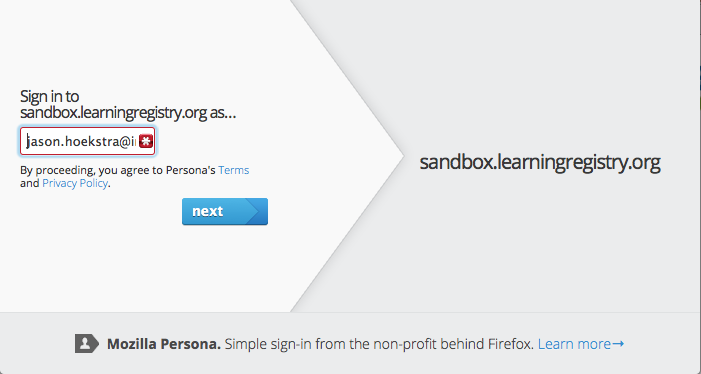
1. **Register for a Digital Signature Username and Password**

Each Learning Registry node offers a service to provide a “digital signature” for each submitted record. A digital signature offers a method to validate the submitter to a known identity as well as allow for update and delete operations in the future. To use the digital signature service, you must create a username and password on the Learning Registry node, then obtain the OAuth keys.

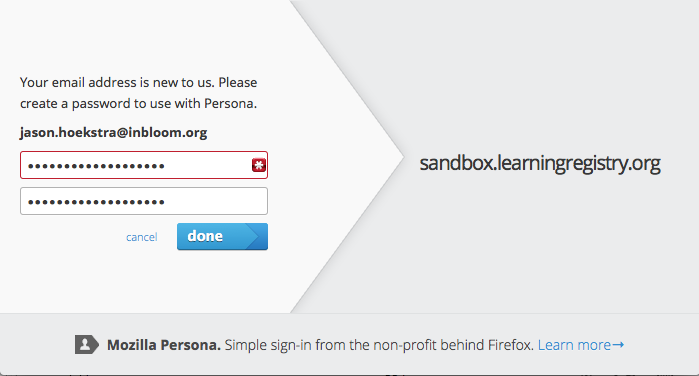
1. Visit the OAuth Key Management service at the intended Learning Registry node: at https://<learning-registry-node-url>/apps/oauth-key-management/ (such as <https://sandbox.learningregistry.org/apps/oauth-key-management/>).



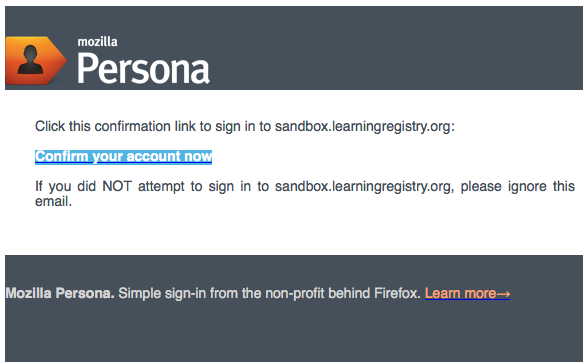
1. The Learning Registry digital signature uses the Mozilla Persona service for authentication. Click on “Sign In”. If you have a Mozilla Persona account, you can skip to Step XX. Otherwise, follow the steps below to create one.
2. Provide your email address:



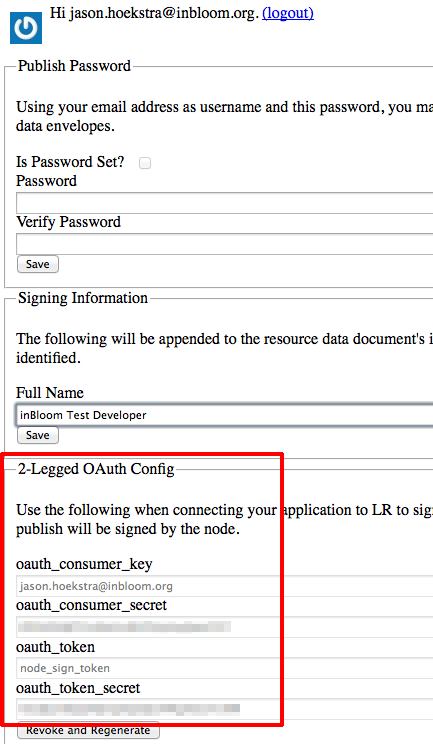
1. Create a password:



1. You will receive a confirmation email, click on the link to confirm your new account.



1. Visit https://<learning-registry-node-url>/apps/oauth-key-management/ again and log in with your Persona ID.
2. Capture the OAuth information as seen below. Keep this in a secure location that you can retrieve in the future, this will be used when you add, update and delete records from the Learning Registry. Capture the oauth\_consumer\_key, oauth\_consumer\_secret, oauth\_token and oauth\_token\_secret fields as highlighted below.



1. **Convert LRMI Metadata from CSV to JSON-LD**

The included “lrmi-csv2jsonld.py” script provides an easy path to convert a LRMI spreadsheet (in CSV format) to JSON-LD required for submission to the Learning Registry. It accommodates both metadata and standards alignment.

1. Open the provided “LRMI-Items-Import-Template.xlsx” spreadsheet in Excel. Notice there are two tabs: Items and Standards. The Items tab provides columns for metadata items and the Standards tab provides columns for standards alignment. To provide standards alignment data, the “url” must match from the Items tab to the “url” of the Standards tab. Multiple standards can be provided for items. Once the Excel spreadsheet is complete, export the Items spreadsheet and Standards spreadsheet separately into CSV format (for example: LRMI-items.csv and LRMI-standards.csv).
2. Run the “lrmi-csv2jsonld.py” script to create the JSON-LD files required to submit to the Learning
   1. First, run the script to generate the item JSON-LD files. “lrmi-csv2jsonld.py –i <items CSV file>” will create one JSON-LD file per row in the CSV spreadsheet. These will be created in a “data/” directory the script will create.
   2. Second, run the script to insert standards into the JSON-LD files. “lrmi-csv2jsonld.py –s <standards CSV file>” will insert the standards into the existing JSON-LD files. Multiple standards will be inserted into the JSON-LD files as associated with the URL from the items file.
   3. Perform a spot check of the JSON-LD files to ensure the LRMI metadata has outputted in the way intended in the spreadsheet.

**Technical Notes from the csv2jsonld-py Script**

Processing Commas in Metadata

If there are commas in the rows metadata, this will result in the creation of arrays in the output. For example:

A row under the column of “learningResourceType” which has the data of “Video,Quiz,Web Page”, will result in:

{ “learningResourceType”: [ “Video”, “Quiz”, “Web Page” ] }

For some columns, this may be undesired behavior. The behavior above can be suppressed with the “-g” option. If your “name” column intentionally has commas as the title of the item, array creation can be skipped. For example, “csv2jsonld.py –i <items CSV file> -g name description” will skip comma array creation for the name and description column.

A row under the column of “name” which has the data of “Abstract Expressionism Pollock's One: Number 31, 1950” and a description which has the data of “This art history video discussion examines Jackson Pollock's \"One: Number 31\", 1950, Oil and enamel paint on unprimed canvas, 1950 (MoMA).” will output as:

{

“name”: “Abstract Expressionism Pollock's One: Number 31, 1950”,

“description”: “This art history video discussion examines Jackson Pollock's \"One: Number 31\", 1950, Oil and enamel paint on unprimed canvas, 1950 (MoMA).”

}

Column Handing in Support of All Schema.org and LRMI Attributes

The items template is not constrained to the columns specified. If additional Schema.org, LRMI attributes or other extensions (such as the [Accessibility Metadata Project](http://www.a11ymetadata.org/)), they can be simply be added to the spreadsheet.

For example, if a “accessMode” column is added with the data “textual,auditory”, the result will be:

{ “accessMode”: [ “textual”, “auditory” ] }

Also, if a JSON object is desired in the output, use the underscores in the columns to output a JSON object.

For example, a column named “publisher\_name” with the data “National Aeronautics and Space Administration” and a column named “publisher\_url” with the data “http://www.nasa.gov” will output:

{

“publisher”:

{

“name”: “National Aeronautics and Space Administration”,

“url”: “http://www.nasa.gov”

}

}

Note: if you add additional attributes outside of the default “CreativeWork” as defined by Schema.org, you may need to extend the @context object as defined starting Line 58 in the csv2jsonld.py script.

1. **Bulk Publish to Learning Registry**

The included “lr-bulk-publish.py” script will submit in bulk JSON-LD files. By utilizing the Learning Registry’s digital signature service, each submission will be automatically signed before inserted into the Learning Registry. This provides publisher authenticity for the submission as well as a method to update and delete records in the future.

This script will first log in to the digital signature service, then bulk publish the JSON-LD files to the Learning Registry. If the log in step fails, the script will halt until resolved.

1. Set constants within the lr-bulk-publish.py script:
   1. On Line 13, set “\_netloc” to the URL of the desired Learning Registry node.
   2. On Lines 22 and 23, set the consumer key and secret obtained from “Register for a Digital Signature Username and Password” step above.
   3. On Lines 27 and 28, set the signing key and secret obtained from the “Register for a Digital Signature Username and Password” step above.
2. Submit a file or files by executing “lr-bulk-publish.py” command. The “-i" parameter specifies the file or files (wildcards are supported) to submit to the Learning Registry.

**Technical Notes from the lr-bulk-publish.py Script**

Automatic Key Insertion from Schema.org Keywords Attribute

Note: if a “keyword” element is present in the LRMI metadata, data from the “keyword” element will also be added to the Learning Registry “keys” attribute in the Learning Registry Envelope. This provides an additional mechanism in the Learning Registry to search for content using keywords.

1. **Verify in the Learning Registry**
   1. Verify the log to ensure records have been submitted successfully. In the log file is the retrieval URL from the Learning Registry. Use this URL to ensure the record is as intended.
   2. It is recommended to use the JSONView browser tool to aide in viewing formatted JSON-LD results in validation. This is available both in [Chrome](https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc?hl=en) and [Firefox](https://addons.mozilla.org/en-us/firefox/addon/jsonview/%E2%80%8E).

## Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Jason Hoekstra | v1.0 | Initial documentation | October 22, 2013 |
| Sue Cowden | v1.1 | Content updates | October 23, 2013 |
| Jason Hoekstra | v1.2 | Corrections to script execution and technical content additions. | October 23, 2013 |

## Planned Additions

* Verify in the Learning Registry Index
* Using the Learning Registry Index
* Update and delete patterns