

Initial Use Case Definitions - Search API Best Practices for Authoritative Data Working Group

Charter	Project Page	Terminology	References	Working Documents	Deliverables	Meeting Notes
-------------------------	------------------------------	-----------------------------	----------------------------	-----------------------------------	------------------------------	-------------------------------

Table of Contents

- [Use Case: User wants to find the URI of an entity from within a metadata editor](#)
 - [Sub-Use Case: User knows keywords related to an entity \(aka Keyword Search\)](#)
 - [Sub-Use Case: User knows the primary label and starts typing it from the first character \(aka Left Anchored Search\)](#)
 - [Cache of entire or significant portion of dataset with updates via retrieve a known concept using a consistent format across datasets](#)
 - [Update cached data](#)
 - [Batch processing](#)
 - [Auto-fill with a batch process](#)
 - [Manual batch processing](#)
 - [Common External Search Format/Ontology](#)
-

Use Case: User wants to find the URI of an entity from within a metadata editor

As a user of a metadata editor application, I want to find an entity in an outside authority to use as metadata in a local record.

Share Key Concepts:

- primary label
 - Authority entities are expected to have a primary label that is a human readable representation of the entity.
- relevant information
 - This will include the primary label of the entity.
 - This likely will include other information from the entity referred to as extended context.
- accurately select
 - This relates to the user's ability to disambiguate similarly labeled entities based on the information provided in search results.

Sub-Use Case: User knows keywords related to an entity (aka Keyword Search)

As a user of a metadata editor application, I want to type in keywords and be presented with a list of relevant information that allows me to accurately select an entity to use as metadata in a local record. The entity is expected to be in the top X (e.g. 5, 8, 10, 20) results, but may be lower in results requiring the ability to access more results.

Sub-Use Case Key Concepts:

- accurately select
 - This assumes that search results will be in rank order with the highest ranked search results appearing first for Keyword Search
- access more results
 - When the desired entity is not in the set of results presented, there needs to be a way for the user to access more results (e.g. pagination)

Sub-Use Case: User knows the primary label and starts typing it from the first character (aka Left Anchored Search)

As a user of a metadata editor application, I want to type in the primary label and be presented with a list of relevant information that allows me to accurately select an entity to use as metadata in a local record. The entity is expected to be the top result in almost all cases.

Sub-Use Case Key Concepts:

- accurately select
 - This assumes that search results will be in alphabetical order for Left Anchored Search

Example: Fill in \$0 MARC field with LOC label search.

Influencing results

- Filter - limit by date ranges, class type, date of birth, language etc.
- Extended Context vs. Filter
- Language filtering can greatly change results
- Fields to search

Cache of entire or significant portion of dataset with updates via retrieve a known concept using a consistent format across datasets

- local search of cached data
- have the URI and want to get details about that term
- get most recent version of data for that URI
- go across multiple authorities
- consistent data access pattern across authorities to get the data

See also update of cached data

Update cached data

- LOC has ATOM feeds that indicate what has changed

Batch processing

Auto-fill with a batch process

- background process that occurs across data

Manual batch processing

- reconciliation through open refine - need to filter by date range

Common External Search Format/Ontology

Requirement: Allow access to vocabulary data via a consistent format, regardless of how the data is managed internally.

Benefits:

- Clear semantics for the fields (e.g. name, birth_date, occupation, etc.) in the format, to be documented only once, rather than by every publisher separately
- Fields provide a common set of data elements to be searched using advanced/fielded search
- Entries from different systems can be easily managed together in a single data management platform, such as a cache or aggregator, without having to re-process the data.
- Rendering and processing code need only be written once and applied to all publishers that provide the common format
- Publishers do not have to change anything they are doing currently to provide another format, only expose an API which transforms their internal data structures into the common format. This can also be done via a third party "shim" that acts as a gateway between the consuming application and the target vocabulary data. (i.e. map from internal representation to external common representation)

[E. Lynette Rayle](#) How to interpret `consistent format`?

Interpretation 1: Format is the encoding language used to express the data (e.g. json, json-ld, rdf-xml, atom, or something else, pick one)

Interpretation 2: Format is the structure of the data (e.g. a person's preferred name can be encoded in a field labeled `preferredName`, `skos:prefLabel`, or something else, pick one)

Interpretation 3: Both the above

[Rob Sanderson](#) Both, somewhat orthogonally. Once the data is structured correctly, it's easy to translate between media types.

The data structure (2) somewhat determines the possibilities for the media type (1).

E.g. skos, in json-ld would be one set of structure + media type.