

LC Working Document - Suggest Service 2.0

The below is a copy-and-pasted document that we (LC) have had floating about internally for whenever we might find the time to redo our Suggest service. It is not final; it has only been informally discussed within our office; it has not been acted on; and there is no guarantee that all the items presented here would be adopted and it is probably safe to assume it would be added to.

It is shared here because of its obvious relationship to the topic under discussion. [Kevin Ford](#) wrote the document so "I" refers to him, and therefore any blame; "us" refers to NDMSO; "QA" is Questioning Authority.

Suggest Service 2.0

Requirements:

1. Search authorized labels, variant labels, deprecated labels, codes, and tokens.
 - a. There are, potentially, diacritic and case-sensitivity issues to consider.
 - b. If a hit is of a variant, deprecated, code or token, then the authorized form needs to be retrieved and displayed possibly with notice. (I'm not wedded to the "display" suggestions; it's just a sample.)
 - i. Example:
 1. Search: Gun do*
 2. Display: Gun dogs (USE: Hunting dogs)
 3. Result: URI for "hunting dogs" is used as is the authorized label
 - ii. Example:
 1. Search: MnU
 2. Display: MnU (USE: University of Minnesota)
2. Example:
 - a. Search: n2009017423
 - b. Display: n2009017423 (USE: Gun Dog (Musical group))
3. Remove duplicates from returned results
4. Search should be left-anchored or, optionally, search anywhere in the label (i.e. not left-anchored)
5. Search params should permit filtering on
 - a. Collections
 - i. This will permit filtering of subdivisions, e.g.
 - b. Schemes
 - i. This could replace the directory search perhaps.
 - c. RDFS types
 - i. Target specific types, such as only Geographics.
6. More sensible result format:
 - a. JSON and XML
 - b. Must/Should include
 - i. Hit quality score
 - ii. Matching label/string
 - iii. Authorized label if matching label/string is not already authorized form.
 - iv. URI
 - c. Might also consider (must balance with performance)
 - i. All variants and codes
 - ii. Sources
 - iii. Relationships
 - iv. And anything else used in the mouseover in the editor

Notes/Questions:

With respect to item 6.c, I think we need to bear in mind users. Everything in 6.c is for integration with our editor but may be of no or limited use to other suggest service users. There may be an additional performance penalty to fetching the additional information that might be OK for us (or mightn't), but otherwise overhead to other users. Just something to bear in mind.

Consider a serialization option that represents QAs format?

Is there anything we like about the current one (and I'm probably thinking specifically of output) that is worth saving?