Solr API

The UCLDC project is running a harvest of objects in collections in both the Nuxeo DAMS and other external sources such as the OAC. In the upcoming releases, we'll be releasing an interface to register your collection for harvest. For now, this collection registry is seeded with previously identified collections. All harvested data is stored in a Solr index in a standardized metadata schema, and can be retrieved using the publicly available Solr API.

The Metadata Schema

The metadata schema was developed to be interoperable with the DPLA metadata schema, while also supporting the needs of the new Calisphere. This schema is still undergoing active development - this page will be updated as changes are made. Subscribe to this page to get an email update when changes are made.

Name	Type	Comments	Multi-Valued
`text`	text_general	not stored; catchall text field for keyword search that indexes tokens - for each object, contains the following fields: `title`, `contributor`, `creator`, `coverage`, `date`, `description`, `extent`, `format`, `identifier`, `language`, `publisher`, `relation`, `rights`, `source`, `subject`, and `type`	yes
`text_rev`	text_general_rev	not stored; the same as the `text` field, but in reverse for efficient leading wildcard queries	yes
`timestamp`	date	timestamp on the Solr document - default value is `NOW`, ie the time of object creation in the Solr index.	no
COLLECTION REGISTRY FIELDS - all multivalued so an object can be related to more than one Campus, Repository, and/or Collection			
`campus`	string	`campus` stores the URL to the registry API campus object	yes
`campus_name`	string	`campus_name` stores the name of the campus, so that clients don't need to look up against the registry API	yes
`collection_url`	string	`collection` stores the URL to the registry API collection object	yes
`collection_name`	string	`collection_name` stores the name of the collection, so that clients don't need to look up against the registry API	yes
`collection_data`	string	`collection_url::collection_name`	yes
`repository_url`	string	`repository` stores the URL to the registry API repository object	yes
`repository_name`	string	`repository_name` stores the name of the repository, so that clients don't need to look up against the registry API	yes
`repository_data`	string	`repository_url::repository_name`	yes
METADATA ON THE METADATA			
`created`	date	refers to creation of the metadata document, not creation of the Solr document, nor creation of the content object	no
`last_modified`	date	refers to the date the metadata document was last modified	no
`created_s`	string	string variant of `created` for wildcard searching	no
`last_modified_s`	string	string variant of `last_modified` for wildcard searching	no
DUBLIN CORE FIELDS			
`title`	text_general	only required field	yes
`contributor`	text_general		yes
`coverage`	text_general		yes
`creator`	text_general		yes
`date`	text_general		yes
`description`	text_general		yes
`extent`	text_general		yes

format	text_general		yes
identifier	text_general		yes
language	text_general		yes
publisher	text_general		yes
relation	text_general		yes
rights	text_general		yes
source	text_general		yes
subject	text_general		yes
type	text_general		yes
date_facet	date		yes
IMAGE FIELDS			
url_item	string	best guess at home url for the item. Filled in by akara? currently indexed to search for items with it filled in, but will likely not be indexed in final release	yes
reference_image_md5	string	not indexed; holds the md5 of the best image found for image objects this will then be passed to the thumbnail server for nicely sized images. For now you can use md5s3stash to calculate url to image	yes
payloads	payloads		yes
version	long		yes
DUBLIN CORE STRING FIELDS - copies of the Dublin Core field by the same name, but stored and indexed as strings, instead of tokenized text			
title_ss	string		yes
contributor_ss	string		yes
coverage_ss	string		yes
creator_ss	string		yes
date_ss	string		yes
description_ss	string		yes
extent_ss	string		yes
format_ss	string		yes
identifer_ss	string		yes
language_ss	string		yes
publisher_ss	string		yes
relation_ss	string		yes
rights_ss	string		yes
source_ss	string		yes
subject_ss	string		yes
type_ss	string		yes
facet_decade	string	https://github.com/ucldc/facet_decade	yes

Coming Soon (finalize by early May)

Name	Type	Comments	Multi-Valued
structmap_url	string	https://github.com/ucldc/ucldc-docs/wiki/media.json	no
structmap_text	string	deep harvest (nuxeo) items only	no
reference_image_dimensions	string	width:height i.e. "100:100", in pixels	no
¿ ga_code ?	string	google analytics code – or, look up from institution-json	no
alternative_title	text_general		yes
genre	text_general		yes
temporal	text_general		yes
rights_holder	text_general		yes
rights_note	text_general		yes
rights_date	text_general		no
provenance	text_general		yes
location	text_general		yes
transcription	text_general		no
		all text_generals also get an _ss string version single valued fields get a *_s string version rights*_date and transcriptio	

Changes for beta launch (finalize by July 24)

Name	Type	Comments	Multi-Valued
sort_collection_data	string	sort_collection_name::collection_name::collection_url	yes
id	string	https://github.com/ucldc/ucldc-docs/wiki/pretty_id	no
harvest_id_s	string	more generic name in anticipation of moving off couchdb. _ss so initially no need to modify schema.	no
sort_date_start	date		no
sort_date_end	date		no
sort_title	string	probably needs some string normalization (remove quotes, initial articles?)	no

schema.xml