Search

User Manual

Solr is a highly configurable search engine with many different options that control its behavior. Solr's default configuration comes with a ton of helpful features which provide a high level of productivity out-of-the-box.

Search Categories

At the time of writing, Solr is not implemented in all areas. However it will be expanded throughout the upcoming releases. The foundation of the integration will allow for basic search functionalities. The following search plugins are available at the time of this writing:

- **Answers**

- **Citations**

- **Collections**

- **Content**

- **Courses**

- **Forum**

- **Groups (regular and super)**

- **Members (public profile and public fields)**

- **Projects**

- **Publications**

- **Resources & Tools**

- **Wiki**

- **Knowledge Base (kb)**

[1] Dependant on Apache Tika installation. This is not a distribution package and must be packaged by a Hubzero package maintainer.

Search Structure

The Hubzero CMS produces a variety of content. For instance you can create a group, a project, a collection, a blog post, a publication, a resource, a tool, a course, a wiki page, a content page, a group page, and so much more. In order to provide a way to search all of the different kinds of data in a consistent manner, a standard model has been developed. You can search a document by title, author, description (which combines all text), path (URL), tags, date, or DOI (if applicable).

If you enter a term "puppy", it will search all fields by default. You can narrow down the search field as described in [Searching Fields](#).

Searching Areas

Search is currently implemented in the site-wide search areas.

There are plans to create an embeddable search module, providing access to the same search index anywhere in the hub. This development has not been completed yet.

Search Result

A search result listing contains a few portions:

1. **Title - The title of the content**

2. **Category - The plugin / facet of the data.**

3. **Date - The date (usually created date) associated with the record.**

4. **Author - Authors listed on the item.**

5. **Tags - Hubzero tags applied to the item.**

6. **Description - A snippet of all matching text in the record.**

7. **Path - The path to the original item.**

Highlighting

A work-in-progress. Marking up information from another system proves to be a little challenging, especially when the data is not sanitized of HTML and other goodies before being indexed. This is implemented at the controller / view level, however it can be handled by Solr itself. There are some system-level CSS and styling issues to work out.

Partial Words

The Hubzero implementation of Solr has not yet been tuned to understand complex phrases in the English language (or any other language for that matter). Therefore Solr sees the words "puppy" and "puppies" as two different concepts. Solr supports partial word matching through the use of the wildcard symbol, *. To perform a search for both "puppies" and "puppy" the query will read "pupp*".

Searching Fields

Solr supports matching within a particular field. This means that a user can specify a term to search for within a particular area of a document.

Searching for title:life, for instance, would return a set of documents containing the word "life".

The following fields are available for filtering, please note the lower-case syntax.

Available Search Fields:

1. **title - filters search down to title results**

2. **author - filters search down to author results**

3. **description - filters search down to full-text results**

4. **path - filters search down to related path results**

5. **doi - filters search by DOI**

* If the object has a DOI issued.

Complex Queries

Solr comes with a query parser which understands Boolean search terms. This means that the keywords "OR" and "AND" can be used. When "OR" is used, Solr returns inclusive results. When "AND" is used, Solr returns exclusive results.

For example, the query "cats OR dogs OR puppies OR pupp" will return any results that contain the words cats, dogs, puppies, or the partial word 'pupp'.

To demonstrate the use of the keyword "AND", the consider the query of "cats AND dogs". This means that Solr will only return the results that have both cats and dogs within the record. In this dataset, this was not a popular combination.

To demonstrate searching for many terms within a given field, consider the query (tags:hacking OR tags:development). This query is interpreted as searching all content which is tagged with either "hacking" or "development.

Let's say you want to search for items tagged with "hacking" or "development" but not tagged with "blog", the query (tags:hacking OR tags:development) AND NOT (tags:blog) will reduce the number of matching results.

More information about the Solr Query Syntax is available on their wiki: https://wiki.apache.org/solr/SolrQuerySyntax**.**