Issue Content Search (Taunton)

The Issue Content Search performs a content search on data, supplied by the publisher, which has been ingested into the Realview database.

The imported data describes what is contained in the digital edition of the publication. Expected data is content title, author, start page, abstract, keywords, subheading etc….

This search interface provides a powerful method for users to search this data to provide relevant results.

Once you have been given a URL and Key by Realview you can view an example/test page at http://yourRealview.url/global/icsearch/searchtest.htm

Search Page

The search page handles the database search. A JSON object describing the search to be performed is POSTed to the page.

URL: http://your.realview.url/global/icsearch/serachcontent.aspx

Search Parameters: A JSON search object is posted to the URL.

```
{
"searchTerm": "",                                    // the term to search for. See Search Syntax.
"pubId": "b7bc29fa-ff82-4088-9597-d480abd5cfe1",    // the identifier of your publication. Contact Realview.
"startDate": "2011-11-08T00:00:00+11:00",           // only search issues published after this data.
"endDate": "2013-11-08T00:00:00+11:00",             // only search issues published before this data.
"deptIds": [ 1 ],                                   // what departments to search. Valid values are returned by /global/icsearch/contentTypes.aspx.
                                                    // you need to specifically tell the service what departments to search.
"resultsPerPage": 10,                               // how many results to return on a page.
"currentPage": 1                                    // what page of the results to return.
"sortBy":0                                          // how to sort the results. 0:Relevence, 1:Date Recent, 2: Date Oldest.
}
```

Return object.

An JSON array of results is returned to the browser for display.

```
{
"thumbnail": "",            // full image URL of the page that contains the hit.
"link": "",                 // full link to page that contains the hit.
"title": "",                // title of the content that contains the hit.
"subheading": "",           // subheading of the content that contains the hit.
"issueDate": "2013-11-08T00:00:00+11:00",  // publish date of the issue that contains the hit.
"issueName": "",            // the issue where the hit is found.
"Abstract": ""              // The abstract of the content that contains the hit.
"text":""                   // text of the article.
"department":""             // name of the department the hit belongs to
"rank":                     // the rank of the result in relation to other results. The higher the more relevant.
"page":                     // the page the item starts on.
}
```

Debugging. If you receive a HTTP Status of 500 you can add &debugon=1 to the query string and the server will write the reason into a Reason header on the HTTP response.

Content Types, Departments.

You need to specify what content types (obj.deptIds) to search when you are posting to the search API.

Valid values for this array are returned by /global/icsearch/contenttypes.aspx

You need to pass your publication key on the query string.

e.g. http://my.realview.url/global/icsearch/contenttype.aspx?pguid=93AD2943-072F-4513-874A-221C5F12DD17

Search Syntax

The search syntax contains a number of operators.

Operator	Description
+, AND	Content must contain this word
-, AND NOT	Content may not contain this word.
|, OR	Content must contain one or the other.
"search term"	Content must contain the exact phrase.

Examples:

Pizza +dough : returns all content containing pizza and dough.

Pizza *AND* dough : returns all content containing pizza and dough.

Pizza +dough –sugar : return all content contain pizza and dough and not sugar.

"pizza dough" : return all content containing the exact phrase "pizza dough".

pizza | dough : returns all content containing pizza or dough.

If no operator is specified then the search assumes an OR operator.

Example:

pizza dough : returns all content containing pizza or dough. Equivalent to *pizza | dough*.

Partial words are also matched.

Example:

Mix : A search for mix will also match mixer, mixed, mixture etc.