Creating Queries for ScienceBase

RESTful Web Service to Query Items Collection

Searching

Searching for items is as simple as sending a GET request with your search fields in the correct parameters. For a basic search in ScienceBase "q" is the search field and "s" must be set to "Search" ("s=Search"). The search is done on http://www.sciencebase.gov/catalog/items. By default HTML is returned, you can return the search as JSON, XML, or ATOM by setting the "format" equal to json, xml, or atom respectively (example: "format=json" for JSON).

Example

Search for "water" returning JSON "http://www.sciencebase.gov/catalog/items?s=Search&q=water"

What is returned

JSON

The returned JSON will have 6 main keys returned: total, rel, url, nextlink, prevlink, and items. “total” will be the number of items returned by the search. "rel" and "url" are interrelated, with "rel" telling you what the "url" is referring to. The "rel" returned in the search results will always be "self" and the "url" is the URL you searched on, thus the rel is telling you that this is a link to itself, ie the link to the page you are on. "nextlink" and "prevlink" will help you navigate through your items, see "Getting more than 20 results" below. "items" is the array of items that were returned from your search. It will always be present, even when empty.

Returned Items

Each item returned will have some properties returned by default: link, id, title, summary, and hasChildren. "link", as above, has the properties of a link to the item: rel and url. The "rel" is "self" and the "url" is the link to the item's html page. "id" is the ScienceBase 2.0 ID. The title and summary are the title and summary attributes of the item. "hasChildren" is boolean and will either be false if the item has no children or true if the item does (and is therefore more of a folder than an item, but can still contain the same properties as a normal item).

Returning more fields

If you need more details about your items returned (and you don't want to go into each item to find these specific properties) you can set the "fields" attribute to whatever you want returned. This overwrites the default returns and leaves you with a base of link and id returned by default. Fields are comma delimited and can be any of the valid fields associated with an item. Fields that are not valid will be silently ignored and fields that are blank will not be returned on individual items.

eg. If I want to return the fields title, facets, summary, and body I would set "fields=title,facets,summary,body" in concert with our water search this would result in the url https://www.sciencebase.gov/catalog/items?s=Search&q=water&format=json&fields=title,facets,summary,body

Advanced Searches and Limiting Searches

There are a number of fields that can be filtered on to give you more precise control of your results. A good way of figuring these out is to use the Advanced Search in ScienceBase. Using the Advanced Search you can actively see the generated search URL that is being created for your search at the bottom of the page next to the Search button. Not everything is visible from this though, so here are a few extra filters you may need:

Searching in a Specific Folder

Set the "folderId" to the identifier of the folder you want to search, for example: "folderId=[sb:id]"

https://www.sciencebase.gov/catalog/items?s=Search&q=water&format=json&folderId=504216b9e4b04b508bfd337d

Returning All Items Cataloged in a Specific Folder

To return all folder items, specify folder id in URL. The folder id serves as the parent for which all child records are retrieved. For example, for GET operation, specify URL: https://www.sciencebase.gov/catalog/items?parentId=5060cc39e4b00fc20c4f3dd7, which will return all metadata records for cataloged notebooks belonging to Missouri Geological Survey collection. "parentId" in URL specifies the return of direct children (1 folder deep). If sub-folders containing their own children exist under the parent, these child items will not be returned in GET requests using parentId. To return all descendants, including child items belonging in sub-folders, "ancestors" must be used: https://www.sciencebase.gov/catalog/items?ancestors=4f4e4761e4b07f02db47dfd0

So if the following folder structure is true: item1 -> item 2 -> item 3

Using "parentID" = item 1, will return item 2. Using "ancestors" = item 1, will return both item 2 and item 3.

To GET only item identifiers, use the following GET operation: https://www.sciencebase.gov/catalog/items?parentId=5060b03ae4b00fc20c4f3c8b&max=1700&format=json&fields=id, a max of 1700 items and the id field is specified for the return response.

Getting more than 20 results

By default a max of 20 items are returned. This can be changed by setting "max" equal to however many you want returned (https://www.sciencebase.gov/catalog/items?parentId=5060b03ae4b00fc20c4f3c8b&max=1700&format=json). However, setting this value very high to return all of the results should be done very cautiously as huge queries will put considerable load on the system. To get around this users can take advantage of the "nextlink" that is provided when more than the max number of items are returned (if less than the max number of responses is returned, or if you are on the last "page" of responses "nextlink" will not be present). The "nextlink" key has two key/value pairs: "rel" which is always "next" for nextlink and "url" which is the URL of the next results page.

Note: Once you are on a page that has pages of results before it, there will also be a key "prevlink" that contains "rel" being equal to "prev" and a URL "url" that is the URL of the page before it.

Using nextlink and prevlink you can traverse through your results.