API Pagination Improvement

I've used a number of other application API's that followed the JSON organizations standard v1.0 (http://jsonapi.org/format/).

There is one component of this specification that I have found particularly useful, and that is the pagination specification (http://jsonapi.org/format/#fetching-pagination)

Having recently used Clowders pagination on search results, I felt it might improve usability to consider adopting the JSON standard approach.

Specifically adding a "links" section like below...

```
"links": {

        "self": "http://example.com/articles",

        "next": "http://example.com/articles?page[offset]=2",

        "last": "http://example.com/articles?page[offset]=10"

}
```

The example uses offsets, but could just as easily been a "from-to" pattern.

The nice thing about this is that it simplifies the approach the consumer must implement to page through the results. They typically make the first call with a "priming" fetch that specifies a number of results to return (eg. page=25) or (eg. From=1&To=25). After that they don't have to keep track of any counters, they can just use the links provided in the result set for further navigation.

If there isn't a "previous" or "first" then you are at the beginning. If there isn't a "next" or "last" then you are at the end. For me, the "self", "first" and "last" are not as useful but I can imagine for some they are.

Update July 2019

Working on a branch to implement results like so.

https://opensource.ncsa.illinois.edu/bitbucket/projects/CATS/repos/clowder/compare/diff?targetBranch=refs%2Fheads%2Finclude-thumbnail-in-search&sourceBranch=refs%2Fheads%2Ffeature%2FCATS-1010-add-pagination-to-search-box-results&targetRepoId=53

http://localhost:9000/clowder/api/search?query=test&size=10&resource_type=file&from=130

```
{
pagination: {
        count: 10,
        size: 10,
        prev: "/api/search?query=test&resource_type=file&from=120&size=10",
        last: "/api/search?query=test&resource_type=file&from=160&size=4",
        from: 130,
        next: "/api/search?query=test&resource_type=file&from=140&size=10",
        first: "/api/search?query=test&resource_type=file&from=0&size=10",
        total_size: 164
},
results: {
        files: [ ... LIST OF 10 FILES OMITTED ],
        datasets: [ ],
        collections: [ ]
}
}
```

...with a pagination subobject describing count of current results, page size, total number of results, page starting index (from), and links for first/last/prev/next.

Results would move over into a results subobject.

Also considering supporting a "page" parameter that users could use with "size" that would auto-calculate "from" based on page*size, but if you provide "from" as well it will prefer that.