Infrastructure - Task #8865

Configure dataone.org web server to redirect DataONE dataset PIRIs

2020-07-02 19:06 - Bryce Mecum

Status:	Closed	**Start date:**	2020-07-01
Priority:	Normal	**Due date:**	
Assignee:	Dave Vieglais	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			
Milestone:	None	**Story Points:**	
Product Version:	*		

Description

As scoped out in https://hpad.dataone.org/8h3o_7VPTlibo5xL9bz24w, we'd like to be able to referenced DataONE resources in a linked-open-data manner. This is useful because it forms the basis of building more interested things on top of them. However, those resources (e.g., Data Packages) don't currently have, subjectively, suitable IRIs, though they have a variety of URLs.

The ideas in the above proposal are multi-tiered but a good first start can be achieved immediately: Support a PIRI space for "Datasets" (DataONE Data Packages) by redirecting requests from their IRI form:

https://dataone.org/datasets/$ID

to their URL form:

https://search.dataone.org/view/$ID.

Such a redirection can be achieved within our Apache configuration using mod_rewrite and a rule similar to:

```
RewriteRule  "^/datasets/(.+)$" "https://search.dataone.org/view/$1" [L,R]
```

History

#1 - 2020-09-15 16:58 - Bryce Mecum

Talked with Dave V and Chris J a few weeks ago and they indicated they wanted to review the Apache config on dataone.org before proceeding. Waiting on that now.

#2 - 2020-09-17 00:31 - Bryce Mecum

I took a look at the config myself and I see no conflicts we can't work around but I do see why caution was warranted. Since launching the website redesign, we put in a few wildcard redirects to keep the old website up (old.dataone.org) and all links resolving as well as keep our Drupal instance available.

The relevant part of the config has rewrites in for three purposes:

1. Enabling access to Drupal

   ```
   ProxyPassMatch ^/(.*\.php(/.*)?)$ "fcgi://127.0.0.1:9000/var/www/www.dataone.org"
   ```

   This doesn't conflict with the proposed config because /foo.php doesn't overlap with /datasets/xyz.

2. Forcing a www subdomain for the main site (i.e., 301 requests like dataone.org to www.dataone.org:

   ```
   RewriteEngine On
   RewriteCond %{HTTP_HOST} !^www\. [NC]
   RewriteCond %{REMOTE_ADDR} !^127\.0\.0\.1
   RewriteRule ^(.*)$ https://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
   ```

This would conflict because our PIRI space is https://dataone.org/datasets/$X so our PIRI redirect needs to be *above* this.

3. Handle links for the old site:

```
RewriteCond /var/www/www.dataone.org%{REQUEST_URI} !-f
RewriteCond /var/www/www.dataone.org%{REQUEST_URI} !-d
RewriteRule ^(.*)$ https://old.dataone.org%{REQUEST_URI} [L]
```

This would conflict too, but we already have to have the PIRI redirect above (2) so this is fine to keep where it is.

So my take on this is that we can put my proposed rule in before or after (1) and we'll be good.

#3 - 2020-09-17 14:49 - Dave Vieglais

Assessment looks good to me.

#4 - 2020-09-17 19:35 - Chris Jones

This looks great Bryce, thanks for closely evaluating the rewrite rules. Matt had mentioned that we may want to transition away from www.dataone.org and primarily use dataone.org. So I think (2) above could probably change such that all links to www.dataone.org get redirected to dataone.org, which also shouldn't affect your addition.

#5 - 2020-09-17 19:41 - Bryce Mecum

Sounds like a great change to me.

#6 - 2020-09-18 03:51 - Bryce Mecum

After testing locally, I went to make this change on dataone.org. Things did not go as planned.

Best I can tell, it turns out that Apache and/or mod_rewrite can't stay away from mangling (encoding/decoding) URLs. I wasn't able to find a combination of directives or mod_rewrite flags that'll simply take exactly what comes after "/datasets/" and redirect to it over on search.dataone.org.

The types of encoded identifiers that cause issues are our doi:10.1234/ABCD and http(s):// identifiers. This is partly to do with having urlencodeable characters but it appears more to do with having slashes in the identifier and encoded slashes in the canonical URL.

The nearest config I got to working was:

```
AllowEncodedSlashes NoDecode
AcceptPathInfo      On
RewriteRule ^/datasets/(.+)$ "https://search.dataone.org/view/$1" [L,R,NE,B=:]
```

NE stops mod_rewrite from re-encoding characters we've already encoded and I came up with B=: because Apache or mod_rewrite can't not decode %3A (:) so we have to re-encode it.

The Holy Grail identifier to test this approach with is https://pasta.lternet.edu/package/metadata/eml/knb-lter-mcm/3104/3 which has a canonical URI of https://dataone.org/datasets/https%3A%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-mcm%2F3104%2F3. Under the above config, this redirect just hard 404s and I have no idea why.

I'm going to continue looking at this tomorrow but if anyone has any hot tips or wants to take a look with me please let me know.

#7 - 2020-10-16 03:04 - Bryce Mecum

I had some time to look at this more closely but haven't come up with a totally satisfying solution. I think I have one though, so read on. Ideally, whatever string of characters is in the path portion of the just simply comes out of Apache unmodified so the downstream client can do what it needs to. Unfortunately, Apache prefers to decode URLs and mod_rewrite appears to get that decoded data and not the raw data. The downstream application does get the raw data but mod_rewrite does ont.

For an example of what I mean by modifying: If we just use a simple RewriteRule, Apache does this for even a relatively easy identifier:

urn%3Auuid%3Ac6feebc4-d822-49a2-860d-32bd808e02f3 -> urn:uuid:c6feebc4-d822-49a2-860d-32bd808e02f3

And what we want is the original input, not the decoded form. I'll note, though:

> URI producing applications should percent-encode data octets that correspond to characters in the reserved set unless these characters are specifically allowed by the URI scheme to represent data in that component. -- https://tools.ietf.org/html/rfc3986#section-2.3

The above implies that : (a reserved character), when in the path part of a URI, doesn't need to be encoded because it doesn't conflict with the interpretation of the URI. So I think technically this is "fine". Just not perfect.

mod_rewrite has a number of flags that I think are relevant here

- [NE] noescape: "By default, special characters, such as & and ?, for example, will be converted to their hexcode equivalent. Using the [NE] flag prevents that from happening."
- [B] escape backreferences: "escape non-alphanumeric characters before applying the transformation."
- [BNP] backrefnoplus (Don't escape space to +)

Note: [NE] doesn't just apply to & and ? but also to % which means we need it on to not re-encode already-percent-encoded chars.

To get a sense of what the options actually look like, I wrote a script to test a few fake and a few real identifiers that exercise our identifier space pretty well. Below is a set of results for various combinations of flags. For each pair of lines, the first line is the input and the second line is the result after Apache + mod_rewrite have had at it.

Note: [L,R] aren't related to encoding but are included for completeness.

[L,R]

```
mypid
mypid

foo%2Fbar%2Cbaz
foo%252Fbar,baz

urn%3Auuid%3Ac6feebc4-d822-49a2-860d-32bd808e02f3
urn:uuid:c6feebc4-d822-49a2-860d-32bd808e02f3

(my*identifier~is'cool)
(my*identifier~is'cool)

doi%3A10.1594%2FPANGAEA.889138
doi:10.1594%252FPANGAEA.889138
https%3A%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-ntl%2F115%2F32
https:%252F%252Fpasta.lternet.edu%252Fpackage%252Fmetadata%252Feml%252Fknb-lter-ntl%252F115%252F32

https%3A%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15%3Fver%3D2018-09-18T03%3A54%3A10.492%2B00%3A00
https:%252F%252Fdoi.org%252F10.5061%252Fdryad.k6gf1tf%252F15?ver=2018-09-18T03:54:10.492+00:00

%7B859BFECB-20E0-483A-9DD7-405DDBCE9052%7D
%7b859BFECB-20E0-483A-9DD7-405DDBCE9052%7d
```

[L,R,NE]

```
mypid
mypid

foo%2Fbar%2Cbaz,"/foo%2Fbar,baz",FALSE
urn%3Auuid%3Ac6feebc4-d822-49a2-860d-32bd808e02f3
urn:uuid:c6feebc4-d822-49a2-860d-32bd808e02f3
```

```
(my*identifier~is'cool)
(my*identifier~is'cool)

doi%3A10.1594%2FPANGAEA.889138
doi:10.1594%2FPANGAEA.889138

https%3A%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-ntl%2F115%2F32
https:%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-ntl%2F115%2F32

https%3A%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15%3Fver%3D2018-09-18T03%3A54%3A10.492%2B00%3A00
https:%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15?ver=2018-09-18T03:54:10.492+00:00

%7B859BFECB-20E0-483A-9DD7-405DDBCE9052%7D
{859BFECB-20E0-483A-9DD7-405DDBCE9052}
```

[L,R,NE,B,BNP]

```
mypid
mypid

foo%2Fbar%2Cbaz
foo%252Fbar%2cbaz

urn%3Auuid%3Ac6feebc4-d822-49a2-860d-32bd808e02f3
urn%3auuid%3ac6feebc4%2dd822%2d49a2%2d860d%2d32bd808e02f3

(my*identifier~is'cool)
%28my%2aidentifier%7eis%27cool%29

doi%3A10.1594%2FPANGAEA.889138
doi%3a10%2e1594%252FPANGAEA%2e889138

https%3A%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-ntl%2F115%2F32
https%3a%252F%252Fpasta%2elternet%2eedu%252Fpackage%252Fmetadata%252Feml%252Fknb%2dlter%2dntl%252F115%252F32

https%3A%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15%3Fver%3D2018-09-18T03%3A54%3A10.492%2B00%3A00
https%3a%252F%252Fdoi%2eorg%252F10%2e5061%252Fdryad%2ek6gf1tf%252F15%3fver%3d2018%2d09%2d18T03%3a54%3a10%2e492
%2b00%3a00

%7B859BFECB-20E0-483A-9DD7-405DDBCE9052%7D
%7b859BFECB%2d20E0%2d483A%2d9DD7%2d405DDBCE9052%7d
```

[L,R,B,BNP]

```
mypid
mypid

foo%2Fbar%2Cbaz
foo%25252Fbar%252cbaz

urn%3Auuid%3Ac6feebc4-d822-49a2-860d-32bd808e02f3
urn%253auuid%253ac6feebc4%252dd822%252d49a2%252d860d%252d32bd808e02f3

(my*identifier~is'cool)
%2528my%252aidentifier%257eis%2527cool%2529

doi%3A10.1594%2FPANGAEA.889138
doi%253a10%252e1594%25252FPANGAEA%252e889138

https%3A%2F%2Fpasta.lternet.edu%2Fpackage%2Fmetadata%2Feml%2Fknb-lter-ntl%2F115%2F32
https%253a%25252F%25252Fpasta%252elternet%252eedu%25252Fpackage%25252Fmetadata%25252Feml%25252Fknb%252dlter%25
2dntl%25252F115%25252F32

https%3A%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15%3Fver%3D2018-09-18T03%3A54%3A10.492%2B00%3A00
https%253a%25252F%25252Fdoi%252eorg%25252F10%252e5061%25252Fdryad%252ek6gf1tf%25252F15%253fver%253d2018%252d09
%252d18T03%253a54%253a10%252e492%252b00%253a00

%7B859BFECB-20E0-483A-9DD7-405DDBCE9052%7D
```

```
%257b859BFECB%252d20E0%252d483A%252d9DD7%252d405DDBCE9052%257d
```

Of all of these, [L,R,NE] looks like the way to go. This is mainly because, while it doesn't stop Apache from mucking with the URL, the result looks equivalent according to the spec and MetacatUI, for example, can handle it. As an example, under this rule, https%3A%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15%3Fver%3D2018-09-18T03%3A54%3A10.492%2B00%3A00 turns into https:%2F%2Fdoi.org%2F10.5061%2Fdryad.k6gf1tf%2F15?ver=2018-09-18T03:54:10.492+00:00 which contains a mix of encoded things and things we'd normally encode such as the ?. That said, when decoded, the result is correct: https://doi.org/10.5061/dryad.k6gf1tf/15?ver=2018-09-18T03:54:10.492+00:00.

I'd love a few sets of eyes on this but I'll make a note to come around to it before too long so we can get this done.

#8 - 2020-11-20 23:59 - Bryce Mecum

- % Done changed from 0 to 100

- Status changed from New to Closed

Added the following to www.dataone.org.conf:

```
# 20201014 mecum
# Enables Dataone /datasets PURI space
# Ref: https://redmine.dataone.org/issues/8865
AllowEncodedSlashes NoDecode
RewriteRule ^/datasets/(.+)$ https://search.dataone.org/view/$1 [L,NE,R]
```

I think this is good enough for now and also as good as we're going to get with a pure-Apache approach.