2017-04-13 DSpace 7 UI Working Group Meeting notes

Date

13 Apr 2017

Angular meeting

Attendees

- Meeting was held in the #angular-ui Slack channel. Anyone could read along
- Full Slack logs at: http://dspace-org.slackarchive.io/angular-ui/ (under April 13, 2017)

Notes

Slack Logs (times are all CDT)

Tim Donohue [9:55 AM]
@here: Reminder that our DSpace 7 UI meetings begin at the top of the hour (~5mins) in Slack. The first 1/2 hour will be Angular UI updates in #angular-ui. The second 1/2 hour will be REST API updates in #rest-api channel.

Tim Donohue [10:01 AM]
@here: Welcome, our DSpace 7 UI meeting begins now in this channel. We'll kick things off with updates on the Angular UI work. As I've been traveling a bit the last few weeks, I'll hand this over to @art-lowel :wink:

Art Lowel [10:01 AM]
hi

Matteo Perelli [10:01 AM]
Hello!

Art Lowel [10:02 AM]
@sourcedump added a PR for the ngrx-store-freeze meta-reducer.
https://github.com/DSpace/dspace-angular/pull/84

[10:02]
ngrx-store-freeze will ensure that the state of the ngrx store isn't mutated anywhere, by applying Object.freeze on it recursively.

[10:02]
More info on object.freeze: https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Object/freeze
Mozilla Developer Network
Object.freeze()
The Object.freeze() method freezes an object: that is, prevents new properties from being added to it; prevents existing properties from being removed; and prevents existing properties, or their enumerability, configurability, or writability, from being changed. The method returns the object being frozen.

[10:02]
That way an exception will be thrown if someone tries to mutate it.

[10:03]
I've reviewed the PR, and it works

Tim Donohue [10:03 AM]
Nice, sounds like something we'd definitely want

Art Lowel [10:03 AM]
Looking at it now, I see that @atarix83 reviewed it as well

[10:03]
So I'll merge it now :slightly_smiling_face:

Waffle APP [10:03 AM]
artlowel moved DSpace/dspace-angular#84 from *Needs Review* to *Done*
#84 ngrx-store-freeze added and configured
Assignees
----
sourcedump

Column

Done

Open in Waffle

Art Lowel [10:04 AM]
I did some more refactoring of the rest services.

[10:04]
I moved requests and responses to their own places in the store, so now the responseCache _only_ handles caching and the state of ongoing request (what their url is, whether we're still waiting for a response, etc.) is tracked in a separate part of the ngrx store.

[10:04]
I'm also making some headway with the loading of related resources, but I'm still having some problems ensuring every object is stored only once in the store.

[10:05]
I've been trying to use the same classes to represent an object in the store and in use in the rest of the application, but it's looking more an more like I'm going to have to let that go, and create a separate version of each model class for use with the serializer and the store.

[10:05]
The problem is that if e.g. an Item has a property `bundles: Bundle[]`. It can't be put in the store as is, because each bundle should be stored separately.

[10:05]
The object cache in the store should only be "one level deep"

[10:05]
While in the store the Item should only contain a reference to where each bundle is located.

[10:05]
So the I started giving every DSO a `self` attribute, and storing bundle objects containing only that `self` attribute in the Item's `bundle` array

[10:05]
But that's problematic when you use it in the front end, because you can't know whether that bundle has been fetched from the rest api already, so instead Item should really have the property `bundles: RemoteData<Bundle>[]`

[10:06]
But since RemoteData objects consist entirely of observables, they can't be serialized from a rest api response.

[10:06]
Hence the conclusion that I'll likely need a set of model classes to represent the normalized form of the data, where every relation is represented by an ID (or an array of IDs), and a set of classes where relations are RemoteData objects.

[10:06]
The former will be used in the ngrx store.

[10:06]
The latter will be used everywhere else, and generated from the normalized versions in the store, using RxJS operators.

Tim Donohue [10:07 AM]
While that does seem like a bit more work, it seems like it also has the benefit of providing a mapping of REST API to model classes, correct? So, it may be less liable to break if the REST API changes slightly

Art Lowel [10:07 AM]
precisely

Tim Donohue [10:08 AM]
So, it seems like a good direction to me then...even though it is a tad more effort (but hopefully not significant work)

[10:09]
Is the plan to move this along into a PR in the nearish future then (I'm assuming this is just on one of your personal branches thus far)

Art Lowel [10:09 AM]
yes, and yes I hope to have a PR before the next meeting

[10:09]
Lotte has been out sick for most of the week, so no further progress to report on the simple Item view page

Tim Donohue [10:10 AM]
oh, that's a shame :slightly_smiling_face: Was going to ask about that. So, that's the first UI "view" being worked on, correct?

Art Lowel [10:10 AM]
indeed

Matteo Perelli [10:11 AM]
I just start to check for the refactoring so I don't have much to say

Tim Donohue [10:11 AM]
It seems like we are nearing the point here where it'll be easier to claim UI "view" tickets and move them forward more rapidly. I see we have a ticket for Collection homepage as well (unclaimed) (edited)

Art Lowel [10:11 AM]
Yeah, that ticket shouldn't be too difficult

[10:11]
If you've already gone through the tutorials on angular.io you should know everything you need to know to get started on that collection homepage.

[10:12]
And I realize we're running low on tasks in the "Ready" column

[10:12]
after I have a PR for the rest relations I'll work on getting some more task descriptions out there

Tim Donohue [10:12 AM]
Do we have a ticket for a Community homepage as well yet?

[10:13]
(And yes, the Ready column could use more tickets: https://waffle.io/DSpace/dspace-angular)

Art Lowel [10:13 AM]
We agreed to remove the concept of communities from dspace 7

Andrea Bollini [10:13 AM]
this is related to the discussion in https://github.com/DSpace/DSpace/pull/1686

Art Lowel [10:13 AM]
was just about to say that :slightly_smiling_face:

[10:14]
so in that ticket andrea asks wether the abstraction should happen in the rest api or in the ui

Andrea Bollini [10:15 AM]
right now merge communities and collections is an idea for the rest api with "low" priority

Tim Donohue [10:15 AM]
I'm not sure I recall this being a finalized decision (maybe my mind is playing tricks on me though). I agree conceptually that this is a "good direction", but I don't know that it's *required* for DSpace 7

Andrea Bollini [10:15 AM]
yes, we have decided to wait after dspace 7 for the real merge (at the database / low api level)

[10:16]
but we also have decided to "sell" communities and collections as merged in dspace 7 from the end-user perspective

Art Lowel [10:16 AM]
That's what I remember as well

Andrea Bollini [10:17 AM]
what happen if the rest api merges these concepts late? is it much work for the angular team to deal with?

Tim Donohue [10:18 AM]
Honestly, here I'd ask how much work we are creating for ourselves. Our timelines are already "tight" (to get something ready / visible by OR2017). So, if this merging complicates either layer, I'd move it towards low priority

Art Lowel [10:18 AM]
ofcourse it both adds and removes work :slightly_smiling_face:

[10:19]
if we have to make separate community routes, components etc…

Andrea Bollini [10:19 AM]
can the store / rest separation help here?

Art Lowel [10:19 AM]
but I think, if the rest api doesn't unify them, we'll end up with a situation where the UI will have to do 2 calls

[10:20]
because for a certain id, you won't know what type of object to expect

[10:20]
so you have to GET both `/communities/:id` and `/collections/:id`

Tim Donohue [10:21 AM]
yes, agreed...two REST calls would be an issue

Andrea Bollini [10:21 AM]
add a READ only endpoint that just make an attempt on the communities and fallback on collection is easy

[10:21]
the complexity is to define a single return object, and manage WRITE

Tim Donohue [10:23 AM]
I guess I'm wondering myself whether it's *easier* to leave Communities as-is (yes, I realize this is more work for Angular team, but the routes /components would be very similar to Collections). Is this too drastic of a change in the midst of building a new UI?

[10:23]
(i.e. we don't have to fix/redo *everything* in DSpace 7, as much as we'd like to)

Art Lowel [10:24 AM]
if you don't think it's feasible at all for dspace 7 @bollini, I agree (to leave communities as they are in dspace 6) (edited)

[10:24]
otherwise the read only fallback will work until OR2017

Andrea Bollini [10:25 AM]
I'm not able to say now for sure that it is feasible... we will know that only after have written some WRITE endpoints

Tim Donohue [10:26 AM]
So, do we want to table this (i.e. delay final decision) for now then and wait on feedback on WRITE endpoints?

Andrea Bollini [10:27 AM]
this could mean to arrive very close to OR

Tim Donohue [10:27 AM]
I admit though, I do worry that there's a lot in the DSpace infrastructure that includes a separation of Communities/Collections (not just the DB layer, but also Solr indexes, statistics, etc). So, I personally have concerns that the REST API will be able to "fake" this merge sufficiently.

Andrea Bollini [10:27 AM]
I will prefer to decide today if we want a read-only single endpoint and the UI to work against that assuming the risk to have two separate endpoint for write

[10:28]
or we leave it out for dspace 7 and also the ui will have communities and collections

Art Lowel [10:29 AM]
As write will have the same problem (if I have a collection object in the UI, I don't know if I have to POST it to `/communities` or`/collections`) I don't think a read only solution will hold up until dspace 8 (edited)

[10:29]
so we're better off leaving it out then (edited)

Tim Donohue [10:30 AM]
By "leaving it out" do you mean we'll do both Communities & Collections in DSpace 7?

Art Lowel [10:30 AM]
indeed

Tim Donohue [10:30 AM]
Ok, I agree. Let's leave Communities & Collections in DSpace 7. The merger will come in DSpace 8

[10:31]
And with that, I realize we are entering into our 2nd 1/2 hour. So, let's move this over to #rest-api to talk REST updates

Waffle APP [10:33 AM]
Art Lowel moved DSpace/dspace-angular#70 from *Needs Review* to *Done*
#70 Add ngrx-store-freeze
Assignees
----
sourcedump

Column

Done

Open in Waffle

Tim Donohue [10:35 AM]
As a final note to @art-lowel... per the discussion above, we probably need a few more tickets in the "Ready" column around Communities. Just a reminder, while I'm thinking about it :wink:

Art Lowel [10:36 AM]
Yeah, I already wrote that down :slightly_smiling_face:

REST meeting

Attendees

- Meeting was held in the #rest-api Slack channel. Anyone could read along
- Full Slack logs at: http://dspace-org.slackarchive.io/rest-api/ (under April 13, 2017)

Notes:

Slack Logs (times are all CDT)

Tim Donohue [10:31 AM]

We are now into the second 1/2 hour of our mtg. I'll hand this over to @bollini to give some #rest-api updates (or @terrywbrady or both) (edited)

Andrea Bollini [10:32 AM]
unfortunately I need to say that I have spent all the past week at the duraspace summit and I'm still recovering from one week off of the office

[10:33]
so just today I'm moving back on DSpace7 staff

Terry Brady [10:33 AM]
Unfortunately, I have been busy with other projects and I have not been able to do much work on REST recently.

Tim Donohue [10:33 AM]
Yes, I wasn't sure who best to provide updates here :slightly_smiling_face:

Andrea Bollini [10:33 AM]
we have a good news

Terry Brady [10:33 AM]
I see that @bollini has provided feedback on a couple of my open PR's

Andrea Bollini [10:33 AM]
@luizsan have join the rest team and I hope we will have more hands (edited)

Tim Donohue [10:34 AM]
Welcome @luizsan!

Luiz Claudio Santos [10:35 AM]
@tdonohue thanks!

Andrea Bollini [10:35 AM]
he has expressed interest in the discovery endpoint

[10:35]
of course we need to work all together to provide more advice

[10:36]
a good place to start *should* be the rest contract https://github.com/DSpace-Labs/Rest7Contract/pull/7

Luiz Claudio Santos [10:36 AM]
Yes, I did a inital commit, but I have to rebase my fork before submit the pool request, I'm gonna to do it in the week...

[10:36]
*pull request

Andrea Bollini [10:36 AM]
it will be useful to have your pull request open anyway

[10:37]
so that we don't forget about that and we can provide comments on your code

Luiz Claudio Santos [10:37 AM]
ok..

Andrea Bollini [10:37 AM]
we will just mark the pr as work in progress until it is not ready / rebased, etc.

Tim Donohue [10:37 AM]
Yes, as mentioned elsewhere, work-in-progress PRs are always welcome. We have a Github label "work in progress" that can be applied (not sure if anyone can apply it or just a committer, but someone can apply on your behalf if needed) (edited)

Luiz Claudio Santos [10:38 AM]
ok, I will do that ASAP.

Andrea Bollini [10:39 AM]
ok, so we will move the discussion over the discovery endpoint implementation on a PR that will be linked from the jira ticket https://jira.duraspace.org/browse/DS-3489

[10:39]
the only new staff that we can "merge" right now is the cited pr on the rest contract from @terrywbrady https://github.com/DSpace-Labs/Rest7Contract/pull/7

[10:40]
he does a great job listing as much as possible endpoints that are needed

Terry Brady [10:40 AM]
Lots of brainstorming in this one. Hopefully with the 4 separate pages we can really flesh out details before implementation

Tim Donohue [10:41 AM]
I agree with @bollini 's note on that PR that we should merge it soon, and open up new issues for specific endpoints

Andrea Bollini [10:41 AM]
yes, and the endpoints page itself should be discussed on many issues / pr

Tim Donohue [10:42 AM]
In general, with the REST contract, I recommend merging PRs quickly (after someone else agrees) as further detailed discussion can be added into new issues/prs

Terry Brady [10:42 AM]
Hopefully these pages can provide a quick overview of the common understanding of our direction

Andrea Bollini [10:43 AM]
I merge the PR #7 now

Tim Donohue [10:43 AM]
as we "stabilize" (or start to finalize) the REST contract, we'll likely change that approach...but currently it's more important to get ideas into the REST contract and clean them up later

[10:44]
Thanks @bollini

Andrea Bollini [10:44 AM]
it could be useful @tdonohue if you add a note in the pr with your last two comments for future reference

Tim Donohue [10:45 AM]
Into the PR you just merged? Or do we want it noted somewhere more generally (e.g. at the top of the README perhaps for now)

Andrea Bollini [10:45 AM]
yes readme is a better place

Tim Donohue [10:46 AM]
I'll update the README later today with a note at the top to detail that this contract is a WIP and changing rapidly (and add my notes mentioned above about quick merger, etc). -> Added this to my TODO

Andrea Bollini [10:46 AM]
latest concrete thing that we can discuss today is
https://github.com/DSpace/DSpace/pull/1686 (edited)

[10:47]
we have already discussed about communities and collections (we keep both in dspace 7)

[10:47]
but in this pr we also discuss about the general behavior that we expect from a rest endpoint

[10:48]
imho /communities should return all the available communities in the system not a subset

Terry Brady [10:48 AM]
The longer that we are able to preserve the existing database structure and concepts from 6 to 7, the easier it will be to have folks participate in testing.

Andrea Bollini [10:49 AM]
if you want only the top communities you need to go /communities/search/top

Mark Wood [10:49 AM]
so for example /communities?depth=1 would give top communities only?

Tim Donohue [10:49 AM]
I agree, `/communities` will likely be expected to return all of them (at least that's how such endpoints normally behave in REST)

Art Lowel [10:49 AM]
That's what I was going to suggest mark

Andrea Bollini [10:50 AM]
/sites/:id/communities will return the top communities as well

Art Lowel [10:50 AM]
the combination of a depth and a scope param seems the most versatile to me

[10:50]
ah, that works as well

Terry Brady [10:51 AM]
Do we need the search if we can get them from the site?

Andrea Bollini [10:51 AM]
the deal here is to provide a general behaviour

Art Lowel [10:51 AM]
so for the direct descendants of a sub-community you'd GET `/communities/:id/communities`?

Luiz Claudio Santos [10:51 AM]
I did the pull request, but I was not able to label it.

Andrea Bollini [10:51 AM]
what should mean scope on items? or bitstreams?

Art Lowel [10:51 AM]
nothing

Andrea Bollini [10:51 AM]
@luizsan give me the link

[10:52]
so the client need to know much more about our api to understand

Luiz Claudio Santos [10:52 AM]
@bollini https://github.com/DSpace/DSpace/pull/1711

Andrea Bollini [10:52 AM]
all our endpoint will always show all the resources

Mark Wood [10:52 AM]
+1 consistency

Terry Brady [10:52 AM]
@luizsan , I added the label

Andrea Bollini [10:52 AM]
all our endpoint that allow subset will have a /search sub-endpoint that will list the available sub-st

Luiz Claudio Santos [10:53 AM]
@terrywbrady tks

Andrea Bollini [10:54 AM]
@art-lowel yes subcommunities are exposed over /communities/:id/communities

Tim Donohue [10:54 AM]
As long as we strive for consistency, and document how you get top-level communities vs child (sub) communities vs all communities, then this sounds good to me

Art Lowel [10:55 AM]
ok @bollini `/site/communities` + `/communities/:id/communities` works for me

Andrea Bollini [10:56 AM]
we have also decided early in the dspace rest 7 discussion that we don't want too deep url

[10:56]
so we will not have /communities/:id/communities/:id/collections etc.

Art Lowel [10:56 AM]
yes ofc, but that's not necessary

Tim Donohue [10:56 AM]
right, you'd just have /communities/:id/collections

Andrea Bollini [10:57 AM]
yes yes, just to recall :slightly_smiling_face:

Tim Donohue [10:57 AM]
Are there any other quick updates to share? (realizing we are nearly out of time)

Andrea Bollini [10:58 AM]
no, I can just share my plan for the next days

[10:58]
I'm going to create initial draft for as much as possible endpoints (registries, eperson, etc.) and ticket to improve the endpoints

[10:59]
we will also setup a public rest 7 endpoint using the data from the official dspace demo

Tim Donohue [11:00 AM]
By the way, should we consider moving the REST7Contract project over to "DSpace" (instead of "DSpace-Labs")? Or do we want to wait until it is more "finalized"?

Terry Brady [11:00 AM]
I will be focused on some projects locally, so feel free to take over any tickets that I have if they are causing a delay. I hope to be able to resume work on the effort in 2-3 weeks.

Andrea Bollini [11:01 AM]
I will prefer to have it on /DSpace now. Just after the warning about the rapid approval of pr is in place

[11:01]
I will get the one about top communities so to finalize our today decision

Tim Donohue [11:02 AM]
:+1: Sounds good, @bollini. I can take on both (hopefully today). I'll move it over to "DSpace/REST7Contract" and update the README with a warning

[11:03]
As we've now hit the top of the hour, we'll close up the official meeting for today. However, anyone is welcome to continue discussion/questions (as always) in either #rest-api or #angular-ui

[11:03]
Thanks all! Next week we'll be on Thurs, April 20 @ 15UTC in Google Hangouts

Tim Donohue [11:52 AM]
@here: I've updated the REST Contract with a README warning (about rapid changes) and moved it to https://github.com/DSpace/Rest7Contract