Arvados - Feature #17119

Virtual folder in FUSE/S3/WebDAV with contents defined by a query

11/16/2020 04:03 PM - Peter Amstutz

Status:	Resolved	**Start date:**	02/12/2021
Priority:	Normal	**Due date:**	
Assigned To:	Ward Vandewege	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2021-03-31 sprint		

Description

Suggestion from customer:

Could we define virtual folders in the S3 / WebDAV / FUSE interface whose contents are defined by a query, e.g. filter on collections that have a certain property?

- Define new group_class called "filter"
- Has arvados filters in a property
- Can use with "group contents" API which reads the filters from the "filter" group properties and applies them to the contents query
- "filter" groups cannot own things and cannot have outgoing permission links
- Add support for showing "filter" groups the same way projects/subprojects are displayed in workbench 2
- Add support for navigating to "filter" groups in Go filesystem (used by keep-web) which uses "group contents" API
- When setting filters there should be some validation to prevent user from setting an obviously invalid filter

Implementation:

In controller, get the group object, check the group_class, if it is "filter" then add the filters to the query, and pass along the query to the backend.

Subtasks:

Task # 17331: Review 17119-make-arvados-path-configurable-in-tests	**Resolved**
Task # 17382: Review 17119-virtual-folder-from-query	**Resolved**
Task # 17430: Review 17119-use-group-contents	**Resolved**
Task # 17473: review 17119-add-filter-groups	**Resolved**
Task # 17479: review 17119-support-filter-groups (workbench2)	**Resolved**

Related issues:

Related to Arvados Epics - Story #16360: Keep-web supports S3 compatible inte...	**Resolved**	**07/01/2020**	**04/30/2021**
Related to Arvados - Feature #17490: [workbench2] make filter groups creatabl...	**New**		

Associated revisions

Revision d8d6bca4 - 03/09/2021 09:44 PM - Ward Vandewege

Merge branch '17119-virtual-folder-from-query'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision f877728c - 03/09/2021 09:45 PM - Ward Vandewege

Merge branch '17119-use-group-contents'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision 949e4651 - 03/12/2021 08:09 PM - Ward Vandewege

Merge branch '17119-controller-bugfix'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision 630d36f4 - 03/12/2021 10:46 PM - Ward Vandewege

Merge branch '17119-make-arvados-path-configurable-in-tests'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision aec10814 - 03/30/2021 08:41 PM - Ward Vandewege

Merge branch '17119-add-filter-groups'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision 9555a202 - 03/30/2021 09:18 PM - Ward Vandewege

Merge branch '17119-support-filter-groups'

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision ebe40163 - 03/31/2021 01:47 PM - Ward Vandewege

In the project/filter group details panel, change the 'Project UUID'
label to just say 'UUID', so that it doesn't cause confusion when
viewing a filter group.

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

Revision f85cd7bb - 03/31/2021 02:24 PM - Ward Vandewege

Fix cypress side panel test now that filter group validation is
stricter.

refs #17119

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <ward@curii.com>

History

#1 - 11/16/2020 04:03 PM - Peter Amstutz

- Status changed from New to In Progress

#2 - 11/16/2020 04:10 PM - Peter Amstutz

- Tracker changed from Bug to Feature

#3 - 11/16/2020 04:10 PM - Peter Amstutz

- Related to Story #16360: Keep-web supports S3 compatible interface added

#4 - 11/16/2020 04:13 PM - Peter Amstutz

- Description updated

#5 - 11/25/2020 07:51 PM - Peter Amstutz

- Description updated

#6 - 11/25/2020 07:58 PM - Peter Amstutz

- Target version set to 2021-01-06 Sprint

#7 - 11/25/2020 08:06 PM - Peter Amstutz

- Status changed from In Progress to New

#8 - 11/25/2020 08:07 PM - Peter Amstutz

- Category set to API

#9 - 12/15/2020 06:13 PM - Peter Amstutz

- Target version changed from 2021-01-06 Sprint to 2021-01-20 Sprint

#10 - 01/06/2021 05:54 PM - Peter Amstutz

- Target version changed from 2021-01-20 Sprint to 2021-02-03 Sprint

#11 - 01/19/2021 07:11 PM - Peter Amstutz

- Description updated

#12 - 01/19/2021 07:11 PM - Peter Amstutz

- Description updated

#13 - 01/19/2021 07:11 PM - Peter Amstutz

- Story points set to 3.0

#14 - 01/20/2021 04:51 PM - Peter Amstutz

- Target version changed from 2021-02-03 Sprint to 2021-02-17 sprint

#15 - 02/03/2021 04:57 PM - Peter Amstutz

- Target version deleted (2021-02-17 sprint)

#16 - 02/03/2021 05:09 PM - Peter Amstutz

- Target version set to 2021-02-17 sprint

- Assigned To set to Ward Vandewege

#17 - 02/08/2021 09:17 PM - Peter Amstutz

FYI I just checked, both arvados.customFileSystem (used by WebDAV/S3) and arvados_fuse.fusedir.ProjectDirectory (FUSE) do separate queries of groups and collections instead of using the "group.contents" endpoint. So they will need to be updated in order for the virtual project behavior to work.

Workbench 2 uses the contents endpoint for project contents and so is likely to work with minimal or no updates.

#18 - 02/08/2021 09:47 PM - Ward Vandewege

- Status changed from New to In Progress

#19 - 02/08/2021 09:48 PM - Ward Vandewege

- Description updated

#20 - 02/12/2021 03:16 PM - Ward Vandewege

I'm splitting this work in multiple chunks:

- add the groups endpoints to Controller => 17119-virtual-folder-from-query
- convert arvados.customFileSystem (used by WebDAV/S3) to use the group contents endpoint => 17119-use-group-contents
- convert arvados_fuse.fusedir.ProjectDirectory (FUSE) to use the group contents endpoint => 17119-use-group-contents
- add the code to handle filter groups to Controller

I have a branch ready for review for the first bullet: 1b79e315b0adf2744e1df1781c198cf698b1c181 on branch 17119-virtual-folder-from-query adds the groups endpoints to Controller. Tests have passed at https://ci.arvados.org/view/Developer/job/developer-run-tests/2330/

#21 - 02/15/2021 11:58 PM - Tom Clegg

17119-virtual-folder-from-query

I think it would be better to add an Included field to the GroupList struct, rather than using a separate SharedGroupList struct. Even though the "shared" endpoint is currently the only one that returns anything there, that won't necessarily be true forever. This would mean "list groups" responses would have "included":null but (other than perhaps some sanity-checking tests which could be updated) I don't think this would bother any clients? The other difference I see is that SharedGroupList doesn't have ItemsAvailable -- is that accidental?

Similarly I think we could add an Include field to arvados.ListOptions, and use that instead of introducing a separate SharedOptions struct. We don't implement it for other APIs yet but it does belong there in principle. (Am I right in thinking the UUID field in SharedOptions isn't really used, i.e., this is the only reason for a separate SharedOptions struct?)

Having a separate ContentsOptions seems a little more appropriate since UUID, Recursive, and ExcludeHomeProject don't make a lot of sense in the generic ListOptions struct. I wonder if it should be called GroupContentsOptions though?

Perhaps EndpointGroupContents2 (groups/{uuid}/contents) should be a little more descriptive like EndpointGroupContentsUUIDInPath. Could have a comment like "alternate HTTP route; client-side code should always use EndpointGroupContents instead"

Can we drop the "Kind" fields? So far, we have it in ContainerRequest but all the other structs live without. It seems like we could avoid adding new ones (and delete it from ContainerRequest?) and instead let lib/controller/router add it to http responses on the fly. (Of course in Go code this field is redundant/useless since each struct has a real type.)

(PipelineInstance)Properties, (PipelineInstance)ComponentsSummary, (PipelineInstance)Components, (PipelineTemplate)Components, (Job)Components, (Job)RuntimeConstraints, (Job)TasksSummary, (Trait)Properties should probably be either map[string]interface{} or (if we don't know whether it will be an object or an array) json.RawMessage.

#22 - 02/17/2021 02:04 PM - Ward Vandewege

- Target version changed from 2021-02-17 sprint to 2021-03-03 sprint

#23 - 02/23/2021 10:47 PM - Ward Vandewege

Tom Clegg wrote:

> 17119-virtual-folder-from-query
>
> I think it would be better to add an Included field to the GroupList struct, rather than using a separate SharedGroupList struct. Even though the "shared" endpoint is currently the only one that returns anything there, that won't necessarily be true forever. This would mean "list groups" responses would have "included":null but (other than perhaps some sanity-checking tests which could be updated) I don't think this would bother any clients? The other difference I see is that SharedGroupList doesn't have ItemsAvailable -- is that accidental?

Sounds good, that is a nice simplification. ItemsAvailable should have been on SharedGroupList, that was an omission (the API server exposes it).

> Similarly I think we could add an Include field to arvados.ListOptions, and use that instead of introducing a separate SharedOptions struct. We don't implement it for other APIs yet but it does belong there in principle.

Done.

> (Am I right in thinking the UUID field in SharedOptions isn't really used, i.e., this is the only reason for a separate SharedOptions struct?)

Yeah, that was a copy/paste from ContentOptions I think, though it was used in `lib/controller/federation/conn.go` which seems dubious - our clients don't send that field afaict. I solved that differently but I am not sure that it's the right approach. I put a 'FIXME' comment in.

> Having a separate ContentsOptions seems a little more appropriate since UUID, Recursive, and ExcludeHomeProject don't make a lot of sense in the generic ListOptions struct. I wonder if it should be called GroupContentsOptions though?

Sure, easy fix, done.

> Perhaps EndpointGroupContents2 (groups/{uuid}/contents) should be a little more descriptive like EndpointGroupContentsUUIDInPath. Could have a comment like "alternate HTTP route; client-side code should always use EndpointGroupContents instead"

Ah, yes, done.

> Can we drop the "Kind" fields? So far, we have it in ContainerRequest but all the other structs live without. It seems like we could avoid adding new ones (and delete it from ContainerRequest?) and instead let lib/controller/router add it to http responses on the fly. (Of course in Go code this field is redundant/useless since each struct has a real type.)

I've done this too.

> (PipelineInstance)Properties, (PipelineInstance)ComponentsSummary, (PipelineInstance)Components, (PipelineTemplate)Components, (Job)Components, (Job)RuntimeConstraints, (Job)TasksSummary, (Trait)Properties should probably be either map[string]interface{} or (if we don't know whether it will be an object or an array) json.RawMessage.

Fixed.

The tests passed at https://ci.arvados.org/view/Developer/job/developer-run-tests/2342/

Ready for another look in a439d4d3228cae6f2d518c24d1d58a259142e5cd on branch 17119-virtual-folder-from-query

#24 - 02/26/2021 04:34 PM - Ward Vandewege

13bddf159a9f39c5d81b5d68402ae8c2f76d0cdb on branch 17119-use-group-contents is ready for review.

#25 - 03/03/2021 04:16 PM - Tom Clegg

17119-virtual-folder-from-query @ a439d4d32...

SharedOptions→ListOptions:

```
+       // FIXME is this right?? We don't have options.UUID to cue the chooseBackend off
+       return conn.chooseBackend(conn.cluster.ClusterID).GroupShared(ctx, options)
```

I think this should be

```
        return conn.chooseBackend(options.ClusterID).GroupShared(ctx, options)
```

> > Can we drop the "Kind" fields?
>
> I've done this too.

Subtle Go trap here, in lib/controller/router/response.go. It's unsafe to append() to a slice that you don't own, like tmp["items"]. If the existing slice's backing array already has enough space, append() will use that instead of creating a new slice, so you might be overwriting stuff that's owned by other code. IIRC the resulting bug was hard to track down when we had one. You can avoid it by starting with a nil slice and appending everything.

(nits) you don't really need to check len(included)>0 -- can just let Go optimize "append nothing"... and since included/includedOK are only used long enough to append, they can be scoped smaller.

```
if included, ok := tmp["included"].([]interface{}); ok {
        items = append(items, included...)
}
```

(nit) renaming itemsOK to isListResponse might make the if-else bit more obvious?

The rest LGTM. Nice readable commits, thanks!

#26 - 03/03/2021 06:12 PM - Peter Amstutz

- Target version changed from 2021-03-03 sprint to 2021-03-17 sprint

#27 - 03/08/2021 10:09 PM - Ward Vandewege

17119-virtual-folder-from-query:

- This branch is poorly named, it should really have been named "17119-add-groups-to-controller" or something like that.
- c869babecb193f02a24f071b8fd101e16aeec680 (https://ci.arvados.org/view/Developer/job/developer-run-tests/2369/)

17119-use-group-contents (branched from 17119-virtual-folder-from-query):

- 13bddf159a9f39c5d81b5d68402ae8c2f76d0cdb (https://ci.arvados.org/view/Developer/job/developer-run-tests/2348/)

17119-add-filter-groups (branched from 17119-use-group-contents):

- Not ready for review yet. (https://ci.arvados.org/view/Developer/job/developer-run-tests/2374/)

#28 - 03/09/2021 02:54 PM - Tom Clegg

17119-use-group-contents LGTM.

#29 - 03/09/2021 03:39 PM - Ward Vandewege

Tom Clegg wrote:

> The rest LGTM. Nice readable commits, thanks!

Thanks for your help with this yesterday, I've made those changes in c869babecb193f02a24f071b8fd101e16aeec680 on branch 17119-virtual-folder-from-query (https://ci.arvados.org/view/Developer/job/developer-run-tests/2369/)

#30 - 03/09/2021 08:57 PM - Tom Clegg

17119-virtual-folder-from-query @ c869babec LGTM -- only nit is that this doesn't really need to be inside the "for _, slice ..." loop:

```
                        if opts.Count == "none" {
                                delete(tmp, "items_available")
```

```
                                    }
```

Harmless at runtime, but maybe worth fixing for clarity.

#31 - 03/09/2021 09:30 PM - Ward Vandewege

Tom Clegg wrote:

> 17119-virtual-folder-from-query @ c869babec LGTM -- only nit is that this doesn't really need to be inside the "for _, slice ..." loop:
>
> [...]
>
> Harmless at runtime, but maybe worth fixing for clarity.

Thanks, I fixed that in [7183ca1596f6509fe2ef1960e8ca948424294781]() and will merge this branch (https://ci.arvados.org/view/Developer/job/developer-run-tests/2377/).

#32 - 03/11/2021 10:08 PM - Ward Vandewege

More branches:

- arvados: 17119-add-filter-groups (not ready for review yet)
- arvados-workbench2: 17119-support-filter-groups (not ready for review yet)
- arvados-workbench2: 17119-make-arvados-path-configurable-in-tests at commit:f21bb210446c4230bef9e5e50b4a9e9b1293bd3b ready for review

#33 - 03/12/2021 09:43 PM - Lucas Di Pentima

Branch 17119-make-arvados-path-configurable-in-tests LGTM, thanks.

#34 - 03/15/2021 01:52 PM - Ward Vandewege

Lucas Di Pentima wrote:

> Branch 17119-make-arvados-path-configurable-in-tests LGTM, thanks.

Thanks, merged.

#35 - 03/16/2021 01:33 AM - Ward Vandewege

arvados: 17119-add-filter-groups ready for review @commit:f61be590b1e2be2b287c3f6df37dd52ad58e6327, tests at https://ci.arvados.org/view/Developer/job/developer-run-tests/2381/

#36 - 03/16/2021 07:49 PM - Tom Clegg

doc/api/methods/groups.html.textile.liquid has "role means FIXME"

In doc/api/permission-model.html.textile.liquid, I think the 2nd sentence here was intended to explain that when an object's owner is a Group, that Group must have group_class="project". So I think this change is incorrect:

```
-Valid uuid types for @owner_uuid@ are "User" and "Group".  For Group, the @group_class@ must be a "project".
+Valid uuid types for @owner_uuid@ are "User" and "Group".  For Group, the @group_class@ must be "filter", "project" or "role".
```

(Perhaps the "For Group" wording could be improved though.)

Maybe this should say "...or a project" instead of "another project" (and "filter group" instead of just "filter"):

```
+* A filter can be owned by a user or another project.
```

Also, the preceding line "A filter cannot own things" → "A filter group cannot own things"

The info in doc/user/topics/projects.html.textile.liquid seems a little out of place here. Most of the user guide is procedural ("here's how to do x" -- like "creating projects" on the "uploading data" page) but much of this page is more like API info.

Perhaps this would be better as a dedicated page ("Using virtual projects"?) showing how to create and use a filter group?

(The discussion about how "home project" and projects in general are implemented, which APIs can be used, etc., seems like it belongs in API docs, not user guide.)

The GroupContents func in lib/controller/rpc/conn.go should really be in localdb, not rpc. ("rpc" is a client-side thing for serializing API calls as http requests -- "localdb" is for API implementation.)

This will panic if respGroup.Properties["filters"] is nil/map/string:

```
+                if filters, ok := respGroup.Properties["filters"]; ok {
+                        for _, f := range filters.([]interface{}) {
```

I think this would fix it:

```
+                if filters, ok := respGroup.Properties["filters"].([]interface{}); ok {
+                        for _, f := range filters {
```

The case where an element of properties.filters is not an array seems like it should be an error, rather than ignore-and-log-warning.

These can panic -- seems worthwhile to use a two-value cast, and return an error if they aren't strings:

```
+                                filter.Attr = tmp[0].(string)
+                                filter.Operator = tmp[1].(string)
```

(...even though none of these things are expected to happen given RailsAPI validation)

Here, "options" is GroupContentsOptions, so it doesn't seem right to pass it to the "get group" endpoint (and passing "select" from the original query doesn't seem right either). Better to use respGroup, err := conn.GroupGet(ctx, arvados.GetOptions{UUID: options.UUID}) (which should become possible when the code moves to localdb).

```
+        var respGroup arvados.Group
+        err := conn.requestAndDecode(ctx, &respGroup, epGet, nil, options)
```

Likewise, once this moves to localdb, "pass through to RailsAPI" code can use something like "return conn.railsProxy.GroupContents(...)" instead of requestAndDecode.

The RailsAPI validation errors say the filters property must be an array of arrays, but it looks like omitting the filters key entirely is equivalent to an "everything" group (no filters at all). I'm inclined to require that to be spelled {"filters":[]} -- that might help avoid things like {"Filters":[["uuid","=","x"]]} being silently ignored.

#37 - 03/17/2021 03:58 PM - Ward Vandewege

- Target version changed from 2021-03-17 sprint to 2021-03-31 sprint

#38 - 03/19/2021 01:32 PM - Ward Vandewege

17119-support-filter-groups on the arvados-workbench2 repository is now ready for review at commit:f288fd05f404b8275974f5f82b738dd354ea8429. Please make sure to run against the tip of 17119-add-filter-groups on the arvados repository to see the functionality.

#39 - 03/19/2021 02:04 PM - Lucas Di Pentima

Ran wb1 test against arvados 17119-add-filter-groups branch at: https://ci.arvados.org/view/Developer/job/developer-tests-workbench2/333/

It seems that all cypress tests are failing because:

- At file src/store/context-menu/context-menu-actions.test.ts
  - Line 137: This overwrites the one I added at line 93. I think it would be better to use the constants, just in case their value change in the future.

The linter doesn't allow duplicated keys on an object. It would be interesting to investigate why the app cannot be built when unit test linting fails, maybe we can add a ticket for that.

#40 - 03/19/2021 02:20 PM - Ward Vandewege

Lucas Di Pentima wrote:

> Ran wb1 test against arvados 17119-add-filter-groups branch at: https://ci.arvados.org/view/Developer/job/developer-tests-workbench2/333/
>
> It seems that all cypress tests are failing because:
>
> - At file src/store/context-menu/context-menu-actions.test.ts
>   - Line 137: This overwrites the one I added at line 93. I think it would be better to use the constants, just in case their value change in the future.
>
> The linter doesn't allow duplicated keys on an object. It would be interesting to investigate why the app cannot be built when unit test linting fails, maybe we can add a ticket for that.

Doh, fixed in commit:7341f64486dd57087f89dd8b4fc92db87c3a9b20

#41 - 03/19/2021 06:19 PM - Lucas Di Pentima

Some more comments:

- At file src/store/context-menu/context-menu-actions.ts
  - Line 208: I believe this kind of checks aren't necessary because of the static type checking that TypeScript offers.
  - Lines 225-235: Indentation used here is 2 spaces instead of 4 as the rest of the codebase. Also, there's some conditional tree nesting that I think could be avoided by AND-ing some checks (I just find shallow conditional trees easier to follow).
- I created a filter group with a non-admin user and it appears under "Projects" on wb2's left side panel, but when I click on it, the breadcrumbs shows "Shared with me > group_uuid", shouldn't be showing the group under "Projects > ...."?
- Project's details panel show filters in an understandable way (not ideal, but usable) but the property editor doesn't know how to edit them (trying to remove one filter or adding one gives an api error), should we fix the property editor or disable the editing features on this kind of groups?
- I think filter groups should have a special icon, to avoid user confusion. Looking at what's available at https://material-ui.com/components/material-icons/, it seems that the one named "Pageview" could a good candidate, to avoid doing cumbersome icon composing.
- When I open the details panel on a filter group, the "Type" label says "Project" instead of "Group".
- When the filter group belongs to a user, it isn't editable (for example, its name cannot be changed). Is that on purpose?

#42 - 03/19/2021 07:53 PM - Ward Vandewege

- Related to Feature #17490: [workbench2] make filter groups creatable/editable added

#43 - 03/19/2021 09:32 PM - Ward Vandewege

Lucas Di Pentima wrote:

> Some more comments:
>
> - At file src/store/context-menu/context-menu-actions.ts
>   - Line 208: I believe this kind of checks aren't necessary because of the static type checking that TypeScript offers.

Interesting - I had added that because the unit tests blew up; router was passed in as undefined. But since then I've added router in the mocked store, so you're right, it is no longer needed. Removed.

> - Lines 225-235: Indentation used here is 2 spaces instead of 4 as the rest of the codebase. Also, there's some conditional tree nesting that I think could be avoided by AND-ing some checks (I just find shallow conditional trees easier to follow).

Indentation fixed (we should probably add a precommit hook?); and I combined the innermost 2 conditions into one.

> - I created a filter group with a non-admin user and it appears under "Projects" on wb2's left side panel, but when I click on it, the breadcrumbs shows "Shared with me > group_uuid", shouldn't be showing the group under "Projects > ...."?

Ah, good catch! That was actually caused by a bug in the API server; filter groups were not being returned in the group contents API call for a user's home project. That's fixed in 29634bb07a1f3c9be44e34b24e71badc4b42a860 in the Arvados repository.

> - Project's details panel show filters in an understandable way (not ideal, but usable) but the property editor doesn't know how to edit them (trying to remove one filter or adding one gives an api error), should we fix the property editor or disable the editing features on this kind of groups?

I've disabled it for now, we'll do a follow on story (#17490) to make these groups editable from wb2.

> - I think filter groups should have a special icon, to avoid user confusion. Looking at what's available at https://material-ui.com/components/material-icons/, it seems that the one named "Pageview" could a good candidate, to avoid doing cumbersome icon composing.

Yes! Change made.

> - When I open the details panel on a filter group, the "Type" label says "Project" instead of "Group".

Yeah, I had added a 'Group class' line underneath but it's better to just change the type, I've made that change.

> - When the filter group belongs to a user, it isn't editable (for example, its name cannot be changed). Is that on purpose?

No, there is no real reason for that; the only thing that matters is that the list of contents of a filter group is not directly editable. I've changed filter groups themselves back to editable.

Ready for review at commit:407d1f609f40e8a7d21cf5846690b63706befaa4 on the 17119-support-filter-groups branch. Tests are at https://ci.arvados.org/job/developer-tests-workbench2/339/. Please note that you need to get the latest commit (16905dd94cb77336351a7a6d2e6ece957f8f3368) on the 17119-add-filter-groups branch (arvados repo) to make all tests pass; I had to make a few more fixes in the controller and API server.

#44 - 03/22/2021 03:26 PM - Lucas Di Pentima

Looking really good! Here're a few additional comments:

- Right-clicking on the filter group gets a different context menu depending whether or not the app is displaying the filter group's contents.
- The DataExplorer component's Type column says 'Project' on filter groups, and the column filtering also doesn't include filter groups as an option. I think this could be added as another follow-up ticket, and not fixed in this one, but just wanted to mention it so we don't forget.
- One "editable project" action that I think should be avoided is 'New project', as no subproject are allowed inside filter groups.

#45 - 03/22/2021 09:42 PM - Ward Vandewege

Lucas Di Pentima wrote:

> Looking really good! Here're a few additional comments:
>
> - Right-clicking on the filter group gets a different context menu depending whether or not the app is displaying the filter group's contents.

Fixed.

> - The DataExplorer component's Type column says 'Project' on filter groups, and the column filtering also doesn't include filter groups as an option. I think this could be added as another follow-up ticket, and not fixed in this one, but just wanted to mention it so we don't forget.

I took care of this, too.

> - One "editable project" action that I think should be avoided is 'New project', as no subproject are allowed inside filter groups.

Fixed.

I also fixed a few more bugs:

- filter groups were not being listed in the side panel trees (except for under the home project)
- trash contents were not listed (fix on the arvados side)

Ready for another look at commit:21abfcbaba4e8e735f353a1e3b030dd5dae8465b on branch 17119-support-filter-groups. You need to run it against the latest commit (840ba2f7cd2d6d73363b966691a07792a324025d) in 17119-add-filter-groups on the arvados repository.

Tests are running at https://ci.arvados.org/job/developer-tests-workbench2/342/

#46 - 03/23/2021 06:17 PM - Lucas Di Pentima

Just one tiny observation :)

- At file src/store/context-menu/context-menu-actions.test.ts - Line 27, inFilterGroup was replaced with readonly. Do you think it would be clearer to name the argument something like forceReadonly or similar? I think renaming it at least on the test case will be useful so that it doesn't get confused with isEditable.

The rest LGTM! thanks.

#47 - 03/23/2021 07:12 PM - Ward Vandewege

Lucas Di Pentima wrote:

> Just one tiny observation :)
>
> - At file src/store/context-menu/context-menu-actions.test.ts - Line 27, inFilterGroup was replaced with readonly. Do you think it would be clearer to name the argument something like forceReadonly or similar? I think renaming it at least on the test case will be useful so that it doesn't get confused with isEditable.
>
> The rest LGTM! thanks.

Good idea! Fixed in commit:c37e5ca83f9bf2dd7f3b1deca62b4ee1614dff99

I'll merge this once the corresponding Arvados branch (17119-add-filter-groups) is merged.

#48 - 03/23/2021 07:47 PM - Ward Vandewege

Tom Clegg wrote:

> doc/api/methods/groups.html.textile.liquid has "role means FIXME"

Ah, whoops, fixed.

> In doc/api/permission-model.html.textile.liquid, I think the 2nd sentence here was intended to explain that when an object's owner is a Group, that Group must have group_class="project". So I think this change is incorrect:
>
> (Perhaps the "For Group" wording could be improved though.)

Indeed! Both fixed.

> Maybe this should say "...or a project" instead of "another project" (and "filter group" instead of just "filter"):
> Also, the preceding line "A filter cannot own things" → "A filter group cannot own things"

All fixed too, thanks.

> The info in doc/user/topics/projects.html.textile.liquid seems a little out of place here. Most of the user guide is procedural ("here's how to do x" -- like "creating projects" on the "uploading data" page) but much of this page is more like API info.
> Perhaps this would be better as a dedicated page ("Using virtual projects"?) showing how to create and use a filter group?
> (The discussion about how "home project" and projects in general are implemented, which APIs can be used, etc., seems like it belongs in API docs, not user guide.)

I agree, it was out of place. I moved it to the API section under "Data management". Is that better?

> The GroupContents func in lib/controller/rpc/conn.go should really be in localdb, not rpc. ("rpc" is a client-side thing for serializing API calls as http requests -- "localdb" is for API implementation.)

Moved, thanks for explaining, I had been wondering about "rpc".

> This will panic if respGroup.Properties["filters"] is nil/map/string:
> I think this would fix it:

Indeed, fixed!

> The case where an element of properties.filters is not an array seems like it should be an error, rather than ignore-and-log-warning.

I made that change.

> These can panic -- seems worthwhile to use a two-value cast, and return an error if they aren't strings:
> (...even though none of these things are expected to happen given RailsAPI validation)

Also converted to comma ok, thanks.

> Here, "options" is GroupContentsOptions, so it doesn't seem right to pass it to the "get group" endpoint (and passing "select" from the original query doesn't seem right either). Better to use respGroup, err := conn.GroupGet(ctx, arvados.GetOptions{UUID: options.UUID}) (which should become possible when the code moves to localdb).

Yes! Fixed.

> Likewise, once this moves to localdb, "pass through to RailsAPI" code can use something like "return conn.railsProxy.GroupContents(...)" instead of requestAndDecode.

Indeed, fixed.

> The RailsAPI validation errors say the filters property must be an array of arrays, but it looks like omitting the filters key entirely is equivalent to an "everything" group (no filters at all). I'm inclined to require that to be spelled {"filters":[]} -- that might help avoid things like {"Filters":[["uuid","=","x"]]} being silently ignored.

Sure, I made that change too.

I also fixed two bugs that were already present in the group contents endpoint before this work:

- items_available:0 was being returned when count: none was requested (doc says it should not be)
- when count: none was requested with a specific offset, too many results were being returned (in a somewhat unpredictable manner)

Ready for review at 056ae0b55536032cadf18d57edcd1c98a9d0ac40, tests passed at https://ci.arvados.org/view/Developer/job/developer-run-tests/2392/

#49 - 03/30/2021 03:24 PM - Tom Clegg

In doc/api/permission-model.html.textile.liquid, "role" should be "filter group":

```
+* A filter group cannot own things (cannot appear in @owner_uuid@).  Putting a role in an @owner_uuid@ field is an error.
```

This would be safer as Index("-j7d0g-")!=5, accounting for the case where someone uses j7d0g as a cluster ID:

```
	if strings.Index(options.UUID, "j7d0g") != 6 {
		return conn.railsProxy.GroupContents(ctx, options)
	}
```

>> The case where an element of properties.filters is not an array seems like it should be an error, rather than ignore-and-log-warning.
>
> I made that change.

OK, "properties":{"filters":[1]} is covered (returns an error), but now I'm noticing that "properties":{"filters":"bogus"} or "properties":{} seems to return all visible items. That should probably cause an error too, right? (That change would also help un-pyramid the "if filters, ok := ...; ok {" block.)

I'm finding the items_available changes in groups_controller pretty hard to follow, primarily because the existing code is so hard to follow. I think I'm getting that when count==none, we still do various things to calculate a value for @items_available, but it's hard to make any sense of what the value actually means, which doesn't matter, because it isn't returned to the caller, because application_controller doesn't copy that instance variable into the response. As long as it works well enough to adjust @offset on each klass iteration, we're ok.

One new mystery is why this has the "|| 0" part -- isn't countless_items_available always a number at this point?

```
        klass_items_available = countless_items_available || 0
```

Come to think of it, isn't this

```
      if params['count'] == 'none'
        # The call to object_list below will not populate :items_available in
        # its response, because count is disabled.  Save @objects length (does
        # not require another db query) so that @offset (if set) is handled
        # correctly.
        countless_items_available = @objects.length
      end
```

```
...
```

```
      if params['count'] != 'none'
        klass_items_available = klass_object_list[:items_available] || 0
      else
        # klass_object_list[:items_available] is not populated
        klass_items_available = countless_items_available || 0
      end
```

...equivalent to this?

```
      klass_items_available = klass_object_list[:items_available] || @objects.length
```

Or maybe it's best to just walk away slowly...

Rest LGTM, thanks.

#50 - 03/30/2021 04:35 PM - Ward Vandewege

Tom Clegg wrote:

> In doc/api/permission-model.html.textile.liquid, "role" should be "filter group":

Oh! Good catch, fixed.

> This would be safer as Index("-j7d0g-")!=5, accounting for the case where someone uses j7d0g as a cluster ID:

Right! Fixed.

>>> The case where an element of properties.filters is not an array seems like it should be an error, rather than ignore-and-log-warning.

> > I made that change.
>
> OK, "properties":{"filters":[1]} is covered (returns an error), but now I'm noticing that "properties":{"filters":"bogus"} or "properties":{} seems to return all visible items. That should probably cause an error too, right? (That change would also help un-pyramid the "if filters, ok := ...; ok {" block.)

Hmm, I'm surprised, those are already caught by the validation afaict:

```
$ arv group create --group '{"group_class":"filter"}'
Error: request failed: http://localhost:8004/arvados/v1/groups: 422 Unprocessable Entity: Properties filters p
roperty missing, it must be an array of arrays, each with 3 elements (req-13otv8ii5ojs21c86zai)

$ arv group create --group '{"group_class":"filter","properties":{}}'
Error: request failed: http://localhost:8004/arvados/v1/groups: 422 Unprocessable Entity: Properties filters p
roperty missing, it must be an array of arrays, each with 3 elements (req-14rceola0knocnbxo9q9)

$ arv group create --group '{"group_class":"filter","properties":{"filters":"bogus"}}'
Error: request failed: http://localhost:8004/arvados/v1/groups: 422 Unprocessable Entity: Properties filters p
roperty must be an array of arrays, each with 3 elements (req-ej0sxzk8y2v1xrtnb3dc)
```

The only one that still works is an empty set of filters, but that seems ok, that's the way to make a "filter group that shows everything":

```
$ arv group create --group '{"group_class":"filter","properties":{"filters":[]}}'
{
 "created_at":"2021-03-30T16:31:39.806186579Z",
 "delete_at":null,
 "description":null,
 "etag":"b1kab85vdbbf09zvopgzeue58",
 "group_class":"filter",
 "href":"/groups/xxxx1-j7d0g-d2za9npzui7nidi",
 "is_trashed":false,
 "kind":"arvados#group",
 "modified_at":"2021-03-30T16:31:39.802918000Z",
 "modified_by_client_uuid":"xxxx1-ozdt8-cx642dwrzaqnoj0",
 "modified_by_user_uuid":"xxxx1-tpzed-etyzf2ixjtvj7fh",
 "name":"xxxx1-j7d0g-d2za9npzui7nidi",
 "owner_uuid":"xxxx1-tpzed-etyzf2ixjtvj7fh",
 "properties":{
  "filters":[]
 },
 "trash_at":null,
 "uuid":"xxxx1-j7d0g-d2za9npzui7nidi",
 "writable_by":[
  "xxxx1-tpzed-etyzf2ixjtvj7fh",
  "xxxx1-tpzed-etyzf2ixjtvj7fh"
 ]
}
```

> I'm finding the items_available changes in groups_controller pretty hard to follow, primarily because the existing code is so hard to follow. I think I'm getting that when count==none, we still do various things to calculate a value for @items_available, but it's hard to make any sense of what the value actually means, which doesn't matter, because it isn't returned to the caller, because application_controller doesn't copy that instance variable into the response. As long as it works well enough to adjust @offset on each klass iteration, we're ok.
>
> One new mystery is why this has the "|| 0" part -- isn't countless_items_available always a number at this point?

Yes that is superfluous, I removed it.

> Come to think of it, isn't this
>
> [...]
>
> ...equivalent to this?
>
> [...]

Not quite - the call to object_list changes @objects, and @objects.length is changed by it. That's why I had to introduce the `countless_items_available` variable.

> Or maybe it's best to just walk away slowly...

Yeah... I'm inclined to do that to avoid introducing new bugs. That function is hairy.

> Rest LGTM, thanks.

Cool - latest @commit:569982eaa34ca549fc25953bcd20570af8e53398 on branch 17119-add-filter-groups

#51 - 03/30/2021 08:36 PM - Ward Vandewege

Ward Vandewege wrote:

> Hmm, I'm surprised, those are already caught by the validation afaict:

I've also added that the extra validation and error in `lib/controller/localdb/conn.go` at a4e59aee7e4f773c72895322a4efbeda3d8f41b8

#52 - 03/30/2021 08:36 PM - Tom Clegg

LGTM, thanks!

#53 - 03/30/2021 09:27 PM - Ward Vandewege

- Status changed from In Progress to Resolved

Tom Clegg wrote:

> LGTM, thanks!

Thanks, everything is merged!

#54 - 05/13/2021 03:10 PM - Peter Amstutz

- Release set to 38