

Arvados - Story #15069

[Workbench 2] Extend search UI to support vocabulary IDs as well as text

04/03/2019 05:38 PM - Tom Morris

Status:	Resolved	Start date:	11/18/2019
Priority:	Normal	Due date:	
Assigned To:	Lucas Di Pentima	% Done:	100%
Category:	Workbench2	Estimated time:	0.00 hour
Target version:			
Description			
See parent ticket for description of modifications needed. This uses the existing APIs and does multiple searches from the client.			
Subtasks:			
Task # 15621: Review 15069-tag-searching-by-ids			Resolved
Related issues:			
Blocks Arvados - Story #14151: Extend vocabulary support for properties to su...			Resolved

Associated revisions

Revision b0bc9cdd - 11/20/2019 09:37 PM - Lucas Di Pentima

Merge branch '15069-tag-searching-by-ids'
Closes #15069

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

History

#1 - 04/03/2019 05:49 PM - Tom Morris

- Parent task deleted (#14151)

#2 - 04/03/2019 05:49 PM - Tom Morris

- Tracker changed from Task to Story
- Description updated

#3 - 04/03/2019 05:49 PM - Tom Morris

- Blocks Story #14151: Extend vocabulary support for properties to support strong identifiers and multiple labels added

#4 - 04/03/2019 05:50 PM - Tom Morris

- Story points set to 3.0

#5 - 04/03/2019 05:56 PM - Tom Morris

- Target version changed from To Be Groomed to Arvados Future Sprints

#6 - 05/29/2019 02:10 PM - Tom Morris

- Subject changed from Extend search UI to support vocabulary IDs as well as text to [Workbench 2] Extend search UI to support vocabulary IDs as well as text

#7 - 05/29/2019 05:02 PM - Eric Biagiotti

- Update WB2 to use the new vocabulary format.
- Text search for properties is available via the search bar at the top. Label search is available in the advanced search view, which populates the search bar automatically.
- When a user starts typing in the advanced search label entries, present an autocomplete list. Display preferred label first.
- When adding a label to a resource, use the same autocomplete view as the advanced search.

#8 - 09/11/2019 03:10 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2019-09-25 Sprint

#9 - 09/11/2019 03:12 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#10 - 09/25/2019 03:00 PM - Lucas Di Pentima

- Target version changed from 2019-09-25 Sprint to 2019-10-09 Sprint

#11 - 09/25/2019 03:20 PM - Tom Morris

- Assigned To deleted (Lucas Di Pentima)

#12 - 09/25/2019 03:35 PM - Tom Morris

- Target version changed from 2019-10-09 Sprint to 2019-10-23 Sprint

#13 - 10/09/2019 03:37 PM - Tom Morris

- Target version changed from 2019-10-23 Sprint to 2019-11-06 Sprint

#14 - 10/23/2019 03:06 PM - Tom Morris

- Target version changed from 2019-11-06 Sprint to 2019-11-20 Sprint

#15 - 11/06/2019 04:46 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#16 - 11/12/2019 07:00 PM - Lucas Di Pentima

- Category set to Workbench2

- Status changed from New to In Progress

- Release set to 22

#17 - 11/18/2019 07:02 PM - Lucas Di Pentima

Updates at commit:366d290 - branch 15069-tag-searching-by-ids (WB2 repo)

The advanced search dialog now autocompletes properties' keys/values using the new vocabulary, and creates the correct search query string. Also does the right thing when arriving from a bookmarked search link, it correctly renders the property "chips" when the advanced search dialog is opened.

Questions for the reviewer

- I'm not sure why I had use <any> on 'const connectVocabularyAndPropertyKey = compose<any>(…)' at src/views-components/resource-properties-form/property-value-field.tsx to be able to pass my own prop to the component. This surely isn't the best solution. Maybe related to this? <https://github.com/DefinitelyTyped/DefinitelyTyped/issues/16990#issuecomment-472885681> (I'm not yet completely used to work with generics)
- The advanced search UI has a preexisting minor bug: when the user adds properties criteria on the search, it keeps adding those properties even if the user enter the same property key twice. The search query string is correctly constructed, though. Should this bug be a blocker for this story? I've been trying to fix it but it doesn't seem trivial for now.

How to test

Just pointing the branch to 4xphq should work, as it would load the new vocabulary example from the branch itself.

#18 - 11/19/2019 04:17 PM - Lucas Di Pentima

I've realized that I was committing with my personal email address. Deleted the branch and re-pushed a corrected one at commit:c33dec5 (wb2 repo)

#19 - 11/19/2019 05:08 PM - Lucas Di Pentima

Updates at commit:2a700288 (wb2 repo)

- Fixes the preexisting bug mentioned on note-17

#20 - 11/19/2019 06:56 PM - Peter Amstutz

(deleted comment, not relevant to this branch)

#21 - 11/19/2019 09:17 PM - Peter Amstutz

I don't know if this is in scope, but if I just put "dog" into the search, it doesn't find anything. However, I can search for "IDVALANIMALS2" and that works. It would be nice if the search recognized "dog" and performed an additional search for "IDVALANIMALS2". Perhaps this is blocked on

[#15070](#) ?

#22 - 11/19/2019 09:28 PM - Peter Amstutz

Related, this seems to assume that all properties have migrated to identifiers, because if you search for "Animal: Dog" then the actual search is has:"IDTAGANIMALS":"IDVALANIMALS2", and you cannot easily search for something that has properties with the literal text "Animal: Dog".

Also, the search history stores (and displays) 'has:"IDTAGANIMALS":"IDVALANIMALS2"' which is not very meaningful to a human going back and choosing a past search.

#23 - 11/19/2019 09:32 PM - Peter Amstutz

It's also possible to have {"IDTAGANIMALS":"IDVALANIMALS2", "Animal: Dog"} for properties on a collection, they will appear the same in the property chips on the collection page.

#24 - 11/20/2019 04:06 PM - Lucas Di Pentima

- Target version changed from 2019-11-20 Sprint to 2019-12-04 Sprint

#25 - 11/20/2019 09:29 PM - Peter Amstutz

- Target version deleted (2019-12-04 Sprint)

Lucas Di Pentima wrote:

Updates at commit:366d290 - branch 15069-tag-searching-by-ids (WB2 repo)

The advanced search dialog now autocompletes properties' keys/values using the new vocabulary, and creates the correct search query string. Also does the right thing when arriving from a bookmarked search link, it correctly renders the property "chips" when the advanced search dialog is opened.

Questions for the reviewer

- I'm not sure why I had use <any> on 'const connectVocabularyAndPropertyKey = compose<any>(…)' at src/views-components/resource-properties-form/property-value-field.tsx to be able to pass my own prop to the component. This surely isn't the best solution. Maybe related to this? <https://github.com/DefinitelyTyped/DefinitelyTyped/issues/16990#issuecomment-472885681> (I'm not yet completely used to work with generics)

I think I figured this one out: commit:17963753e57340121942857300c41c7852275c4b

As we discussed at the sprint meeting, since it requires API server support for OR queries and there's also some differences of opinion about what the right search UI would be, the comment on note-21 about making free text search aware of identifiers/aliases is out of scope.

Rest LGTM, thanks!

#26 - 11/20/2019 09:52 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved