Arvados - Feature #13561

[API] Store, and add APIs to retrieve, previous versions of collection objects

06/04/2018 03:58 PM - Tom Clegg

Status:	Resolved	**Start date:**	10/04/2018
Priority:	Normal	**Due date:**	
Assigned To:	Lucas Di Pentima	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2018-10-31 sprint		

Description

API described at Collection version history

Subtasks:

Task # 14170: Review 13561-collection-versions-api	**Resolved**
Task # 14297: Update documentation	**Resolved**
Task # 14298: Review documentation: 13561-collection-versions-doc	**Resolved**
Task # 14375: Review 13561-collection-versions-fixes	**Resolved**
Task # 14379: Review 13561-trashed-collection-versions-fix	**Resolved**

Related issues:

Related to Arvados - Feature #13109: Support collection versions	**Resolved**	
Related to Arvados - Story #14299: [keep-balance] Ensure blocks referenced by...	**Resolved**	**10/23/2018**
Related to Arvados - Bug #15148: keep-balance incorrectly accounts for blocks...	**Resolved**	**04/26/2019**
Blocks Arvados - Story #13494: Browse previous versions of a collection	**Resolved**	**02/19/2020**

Associated revisions

Revision a31587cb - 10/11/2018 06:36 PM - Lucas Di Pentima

Merge branch '13561-collection-versions-api'
Refs #13561

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

Revision 34c144c1 - 10/17/2018 08:12 PM - Lucas Di Pentima

Merge branch '13561-collection-versions-fixes'
Refs #13561

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

Revision d63756d4 - 10/22/2018 04:30 PM - Lucas Di Pentima

Merge branch '13561-trashed-collection-versions-fix'
Refs #13561

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

Revision 0adb096a - 10/23/2018 01:56 PM - Lucas Di Pentima

Merge branch '13561-collection-versions-doc'
Closes #13561

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

History

#1 - 06/04/2018 03:58 PM - Tom Clegg

- Tracker changed from Bug to Feature

#2 - 06/04/2018 04:40 PM - Tom Morris

- Blocks Story #13494: Browse previous versions of a collection added

#3 - 06/04/2018 04:40 PM - Tom Clegg

- Related to Feature #13109: Support collection versions added

#4 - 07/11/2018 02:45 PM - Tom Morris

- Target version set to To Be Groomed

#5 - 08/08/2018 04:40 PM - Tom Clegg

- Subject changed from [API] Retrieve previous versions of collection objects to [API] Store, and add APIs to retrieve, previous versions of collection objects

#6 - 08/08/2018 04:43 PM - Tom Morris

- Story points set to 4.0

#7 - 08/21/2018 05:56 PM - Tom Morris

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

#8 - 09/05/2018 03:50 PM - Tom Morris

- Assigned To set to Lucas Di Pentima

- Target version changed from Arvados Future Sprints to 2018-09-19 Sprint

#9 - 09/06/2018 07:26 PM - Lucas Di Pentima

- Status changed from New to In Progress

#10 - 09/19/2018 03:06 PM - Lucas Di Pentima

- Target version changed from 2018-09-19 Sprint to 2018-10-03 Sprint

#11 - 09/19/2018 03:14 PM - Lucas Di Pentima

- Target version changed from 2018-10-03 Sprint to 2018-09-19 Sprint

#12 - 09/19/2018 03:15 PM - Lucas Di Pentima

- Target version changed from 2018-09-19 Sprint to 2018-10-03 Sprint

#13 - 10/03/2018 03:17 PM - Lucas Di Pentima

- Target version changed from 2018-10-03 Sprint to 2018-10-17 sprint

#14 - 10/03/2018 04:09 PM - Tom Clegg

- Related to Story #14299: [keep-balance] Ensure blocks referenced by old collection versions are safe from garbage collection added

#15 - 10/04/2018 08:09 PM - Lucas Di Pentima

Initial implementation at a18fe628853e2042bb104088dd586cb8f41adcef - branch 13561-collection-versions-api
Test run: https://ci.curoverse.com/job/developer-run-tests/920/

- Collection model creates versions when being updated, depending of site configuration parameters.
- Old versions can't be updated. They have selected fields synchronized with the current version' values.
- Links can't be attached to old versions.
- Collections.index API expanded to include past versions.
- Tests added

(Some wb integration tests failed -- investigating)

#16 - 10/04/2018 09:58 PM - Lucas Di Pentima

Updates at 652e26fee
Test run: https://ci.curoverse.com/job/developer-run-tests/921/

- Ignore version & current_version_uuid attributes at create API endpoint
- Add include_old_versions param on discovery doc

#17 - 10/05/2018 02:47 AM - Lucas Di Pentima

Update at 728b7ee1f
Test run: https://ci.curoverse.com/job/developer-run-tests/922/

Realized one key feature was missing: Allow old collection versions to be accessible by UUID.

#18 - 10/05/2018 02:08 PM - Lucas Di Pentima

Merged master into the branch at 0f11eb78e
Test run: https://ci.curoverse.com/job/developer-run-tests/924/

#19 - 10/05/2018 06:57 PM - Lucas Di Pentima

Documentation updates at 9d8a391d4 - branch 13561-collection-versions-doc

Ready for review

#20 - 10/09/2018 04:57 PM - Tom Clegg

Implementing this by overriding save! seems sketchy: AFAIK neither update_attributes nor save call save! (rather, save! is implemented as save || raise). Can we rephrase this as a Rails callback -- perhaps after_save?

I find it hard to follow the copying of attributes between self and snapshot. I wonder if we should either use @old_attributes from log_start_state, or use SQL directly (after getting the row lock) to copy the current version to a new row with a new UUID.

A unit test like this might be useful to detect loopholes:

```
c = collections(...)
c.name = 'foobar'
c.save
c.properties['foo'] = 'bar'
c.update_attributes(name: 'baz')
```

The tests are in test/functional but they don't call the controllers -- seems like they should move to test/unit?

"older versions should no(sic) be directly updatable" should use an admin token.

#21 - 10/09/2018 09:37 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Documentation updates at 9d8a391d4 - branch 13561-collection-versions-doc

In "Using collection versioning"

- Suggest rephrasing first section:

  When collection versioning is enabled, updating certain collection attributes (name, description, properties and manifest_text) will save a copy of the previous collection version...

- Need to explain versioning more clearly: the past version is saved with a new uuid, current_version_uuid points to the most recent version, the version number is incremented.
- Cut out the extra fields in the "arv" examples, limit it to "uuid", "current_version_uuid", "version", "created_at", "modified_at"
- Adjust heading: "Manually preserving a version" → "Ensuring a version is preserved"

#22 - 10/10/2018 01:24 AM - Lucas Di Pentima

Updates at 1499536da
Test run: https://ci.curoverse.com/job/developer-run-tests/926/

Tom Clegg wrote:

> Implementing this by overriding save! seems sketchy: AFAIK neither update_attributes nor save call save! (rather, save! is implemented as save || raise). Can we rephrase this as a Rails callback -- perhaps after_save?
> I find it hard to follow the copying of attributes between self and snapshot. I wonder if we should either use @old_attributes from log_start_state, or use SQL directly (after getting the row lock) to copy the current version to a new row with a new UUID.

Refactored the code so that's hopefully more readable, and also to avoid overriding save!(), by taking advantage of around_update and after_update callbacks.

> "older versions should no(sic) be directly updatable" should use an admin token.

Fixed

#23 - 10/10/2018 08:00 PM - Tom Clegg

AFAICT after_update runs after around_update is done, which seems to mean sync_past_versions is outside the "with_lock" block. There's also a potential race where should_preserve_version? returns false early in the process, then starts returning true later on because time has advanced. Would it be simpler to do all of the steps in around_update? Something like

```
with_lock
  return(yield) unless { ... need snapshot ... }

  snapshot = ...
  yield
  snapshot.save
end
```

Since old_versions_cannot_be_updated is a validation, I think it should call errors.add() and return false, rather than raising PermissionDeniedError.

Should test that current_version_uuid is ignored during update (create is already tested)

#24 - 10/10/2018 08:52 PM - Lucas Di Pentima

Peter Amstutz wrote:

> In "Using collection versioning"
>
> - Suggest rephrasing first section:
>
>   When collection versioning is enabled, updating certain collection attributes (name, description, properties and manifest_text) will save a copy of the previous collection version...

[...]

> - Adjust heading: "Manually preserving a version" → "Ensuring a version is preserved"

Suggestions addressed at dff1e79
Re-phrased/re-arranged user guide. Added admin guide.

#25 - 10/11/2018 04:22 PM - Lucas Di Pentima

Tom Clegg wrote:

> AFAICT after_update runs after around_update is done, which seems to mean sync_past_versions is outside the "with_lock" block. There's also a potential race where should_preserve_version? returns false early in the process, then starts returning true later on because time has advanced. Would it be simpler to do all of the steps in around_update? Something like

Oops, sorry... I guess I misunderstood the docs when checking that. Nevertheless, this worked because with_lock uses a "SELECT ... FOR UPDATE" lock that's released after COMMIT, but I agree with you that's better to be explicit about it.

> [...]
> Should test that current_version_uuid is ignored during update (create is already tested)

All comments addressed at c8b76b1d7
Test run: https://ci.curoverse.com/job/developer-run-tests/927/

#26 - 10/11/2018 06:27 PM - Tom Clegg

LGTM @ c8b76b1d7, thanks!

#27 - 10/15/2018 07:13 PM - Peter Amstutz

Reviewing 13561-collection-versions-doc @ dff1e793412d9fc673c0158149db812649214399

Overall this writeup is much better, thank you.

- Formatting comment: in a few places you have a line break mid-paragraph, which is awkward. You need to either join those into one long line in textile, or add an extra space so they are separate. Example:

  Every collection has a version attribute that indicates its version number, starting from 1 on new collections and incrementing by 1 with every versionable update. All collections point to their most current version via the current_version_uuid attribute, being uuid and current_version_uuid equal on those collection records that are the the current version of themselves.
  Note that the "current version" collection record doesn't change its uuid, "past versions" are saved as new records every time it's needed, pointing to the current collection record.

- Needs discussion about what happens if I delete a collection with versioning enabled, or with preserve_version = true.

- In your example:

```
arv collection index --filters '[["uuid", "=", "o967z-4zz18-ynmlhyjbg1arnr2"]]' --include-old-versions
```

The filter is ["uuid", "=", "o967z-4zz18-ynmlhyjbg1arnr2"] but it also returns one or more collection with a different uuid, which don't actually fit the filter. That seems really weird. Is that really how it works or is there a typo? If this is a special case, you need to elaborate more on how it works. On the design side, was there any discussion of using the "included" response field to hold past versions?

- Might want to provide an example of filtering on "current_version_uuid" and "version" to get a stable version even as new versions are created.

- Is there a way to request a particular version as part of a GET request?

#28 - 10/17/2018 03:19 PM - Lucas Di Pentima

- Target version changed from 2018-10-17 sprint to 2018-10-31 sprint

- Story points changed from 4.0 to 0.5

#29 - 10/17/2018 06:57 PM - Lucas Di Pentima

After-demo fixes on 9fd178164 - branch 13561-collection-versions-fixes
Test run: https://ci.curoverse.com/job/developer-run-tests/937/

- Request individual collections with its past versions using [["current_version_uuid", "=", "some-uuid"]] instead of filtering by UUID, to avoid confusing behavior.
- Maintain consistency on illegal update attempts by raising an exception when trying to change version or set preserve_version to false when already set to true.

#30 - 10/17/2018 07:08 PM - Tom Clegg

I think the spec and previous version were right about ignoring preserve_version=false, rather than raising an error. Since there is no case where passing preserve_version=false could *prevent* a version from being preserved, the flag means "don't *require* this version to be preserved". It doesn't mean "un-preserve".

Filter changes LGTM.

#31 - 10/17/2018 07:31 PM - Lucas Di Pentima

Reverted requested changes & rebased at eab1bd7
Test run: https://ci.curoverse.com/job/developer-run-tests/938/

Waiting for the tests to pass before merging.

#32 - 10/18/2018 10:51 PM - Lucas Di Pentima

While testing things for the documentation branch, came across a bug that made trashed past versions to appear on index API calls when being an admin and passing include_trash.

Fixed at 89acce74a (branch 13561-trashed-collection-versions-fix) and added new tests.
Test run: https://ci.curoverse.com/job/developer-run-tests/942/

#33 - 10/22/2018 03:29 PM - Lucas Di Pentima

Peter Amstutz wrote:

> - Formatting comment: in a few places you have a line break mid-paragraph, which is awkward. You need to either join those into one long line in textile, or add an extra space so they are separate.

Done

> - Needs discussion about what happens if I delete a collection with versioning enabled, or with preserve_version = true.

Added a note specific to this topic, I hope it's enough.

> - In your example:
>   [...]
>   The filter is ["uuid", "=", "o967z-4zz18-ynmlhyjbg1arnr2"] but it also returns one or more collection with a different uuid, which don't actually fit the filter. That seems really weird. Is that really how it works or is there a typo? If this is a special case, you need to elaborate more on how it works. On the design side, was there any discussion of using the "included" response field to hold past versions?

Fixed on the other branch, updated the doc accordingly.

> - Might want to provide an example of filtering on "current_version_uuid" and "version" to get a stable version even as new versions are created.

Added example.

> - Is there a way to request a particular version as part of a GET request?

Right now, you can do a GET by UUID to past versions (there's an example about that). Didn't add a specific call like /collections/UUID/VerNum to the API. Should I do it?

Rebased branch 13561-collection-versions-doc - e5dc0c9de

#34 - 10/22/2018 03:53 PM - Tom Clegg

13561-trashed-collection-versions-fix @ 89acce74a LGTM, thanks

#35 - 10/23/2018 01:47 PM - Peter Amstutz

This looks good, just a few suggested edits, then please merge:

> Note that if you set collection_versioning to false after being enabled, old versions that could have been saved will still be accessible.

The second half of this sentence is confusing. Suggest: "old versions will still be accessible, but further changes will not be versioned."

> There are two ways that this feature works:

Suggest: "A version will be saved when one of the following conditions is true:"

> When a collection’s current version is deleted, its past versions will follow it.

Suggest: "When a collection is deleted, any past versions are deleted along with it."

#36 - 10/23/2018 02:16 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved

Applied in changeset arvados|0adb096afc227db376823f84956de6d7ea30dc10.

#37 - 11/13/2018 08:53 PM - Tom Morris

- Release set to 14

#38 - 09/23/2019 02:55 PM - Tom Morris

- Related to Bug #15148: keep-balance incorrectly accounts for blocks in collections with null `modified_at` field added