Arvados - Story #13111

[WebDAV] Support browsing of project hierarchies

02/21/2018 06:49 PM - Tom Morris

Status:	Resolved	**Start date:**	03/27/2018
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2018-05-09 Sprint		

Description

Expose tree of project/collection names, e.g., keep-web-host/users/bob/projectname/collectionname/foo.txt

This is an initial read-only implementation that exposes the /users/*/ hierarchy with some limitations. There will be followup work (write access, shared projects, better performance), see #13272.

Subtasks:

Task # 13225: Review 13111-projects-by-id	**Resolved**
Task # 13416: Review 13111-no-anonymous-sitefs	**Resolved**

Related issues:

Related to Arvados - Story #13146: [API] Endpoint to get projects shared with me	**Resolved**	**08/15/2018**
Related to Arvados - Story #13218: Support browsing of projects shared with m...	**New**	
Blocks Arvados - Feature #13272: [keep-web] read/write support for /by_id and...	**New**	

Associated revisions

Revision 8da66eb4 - 04/20/2018 07:59 PM - Tom Clegg

Merge branch '13111-webdav-projects'

refs #13111

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

Revision d907e7fe - 04/24/2018 03:07 PM - Tom Clegg

Update mock to match new arvados.File interface.

refs #13111

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

Revision fcfbbddf - 04/26/2018 03:42 PM - Tom Clegg

Merge branch '13111-projects-by-id'

refs #13111

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

Revision dc0dae1a - 05/01/2018 06:06 PM - Tom Clegg

Merge branch '13111-no-anonymous-sitefs'

refs #13111

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

History

#1 - 02/28/2018 09:04 PM - Tom Morris

- Parent task deleted (#13110)

#2 - 02/28/2018 09:07 PM - Tom Morris

- Tracker changed from Task to Story

- Description updated

#3 - 02/28/2018 09:08 PM - Tom Clegg

A good portion of this is currently sitting on an unmerged #12308 branch.

#4 - 03/07/2018 08:23 PM - Peter Amstutz

- Related to Story #13146: [API] Endpoint to get projects shared with me added

#5 - 03/07/2018 08:44 PM - Tom Morris

- Story points set to 1.0

#6 - 03/14/2018 07:26 PM - Tom Morris

- Description updated

- Assigned To set to Tom Clegg

- Target version changed from To Be Groomed to 2018-03-28 Sprint

#7 - 03/14/2018 07:28 PM - Tom Morris

- Related to Story #13218: Support browsing of projects shared with me in WebDAV added

#8 - 03/19/2018 06:37 PM - Tom Clegg

- Status changed from New to In Progress

#9 - 03/19/2018 07:08 PM - Tom Clegg

Will it be too confusing if https://keep-web-host/home/$collectionname serves different content depending on the auth token? It certainly seems to violate the spirit of URIs.

Other possibilities:

- https://keep-web-host/home/$collectionname → 307 redirect to https://keep-web-host/by_id/$userUUID/$collectionname (how many webdav clients will follow this?)
- https://keep-web-host/users/$username/$collectionname (don't support https://keep-web-host/home/$collectionname at all)
- https://$username.keep-web-domain/home/$collectionname

https://keep-web-host/users/$username/$collectionname has an advantage that it can map directly to FUSE as $keepmount/users/$username/$collectionname

#10 - 03/23/2018 05:40 PM - Tom Clegg

- Description updated

#11 - 03/23/2018 06:28 PM - Tom Morris

I believe that we agreed that

```
https://keep-web-host/users/$username/$collectionname
```

is the way to go here.

#12 - 03/23/2018 07:49 PM - Tom Clegg

- Blocks Feature #13272: [keep-web] read/write support for /by_id and /users added

#13 - 03/23/2018 07:54 PM - Tom Clegg

- Description updated

#14 - 03/27/2018 01:26 PM - Tom Clegg

From discussion offline: we're aiming to get the "projects tree" part of the Go SDK merged and exposed through keep-web with some simplifying restrictions. Some real-world experience is likely to help guide further development (especially performance); besides, doing the whole thing at once would mean reviewing an unnecessarily large chunk.

13111-webdav-projects @ 127c8584fa1a2336fd6569d9d6013f3f2b166088https://ci.curoverse.com/job/developer-run-tests/668/

Specifically:

- /users tree is read-only

- no performance tuning (in particular, no cache)

#15 - 03/28/2018 03:20 PM - Tom Clegg

- Target version changed from 2018-03-28 Sprint to 2018-04-11 Sprint

#16 - 03/29/2018 02:52 PM - Peter Amstutz

- is there a sensible way to split the branch so that the go FUSE stuff isn't getting merged along with the webdav changes?

- can we stop giving things confusing names? like crunchrunner vs crunch-run and now arvadosclient vs arvados-client

#17 - 03/29/2018 03:02 PM - Peter Amstutz

I don't know if this was on your master plan, but apparently cgofuse supports WinFSP, which means a future Go FUSE driver could run natively on Windows, as an alternative to our current plan to use Windows' WebDAV support?

#18 - 03/29/2018 03:35 PM - Peter Amstutz

Apologies if some of these comments arn't directly related to the branch, the diff is very large and includes refactoring of existing code (which appears as additions in the patch) so I'm wandering around a bit trying to get a handle on it.

```
// A File is an *os.File-like interface for reading and writing files
// in a FileSystem.
type File interface {
    io.Reader
    io.Writer
    io.Closer
    io.Seeker
    Size() int64
    Readdir(int) ([]os.FileInfo, error)
    Stat() (os.FileInfo, error)
    Truncate(int64) error
    Sync() error
}
```

This comment need more information.

- "reading and writing files in a FileSystem." but presumably a given instance only refers to one file
- Since it has Readdir, It can also represent a directory?
- it's more like a filehandle, and the underlying file data and metadata is represented by an inode?

```
    // TODO: If the nearest common ancestor ("nca") of olddirf and
    // newdirf is on a different filesystem than fs, we should
    // call nca.FS().Rename() instead of proceeding. Until then
    // it's awkward for filesystems to implement their own Rename
    // methods effectively: the only one that runs is the one on
    // the root filesystem exposed to the caller (webdav, fuse,
    // etc).
```

This is a confusing, is this talking about separate file system mounts (from the kernel POV) or Filesystem objects within the Go program? Does this mean moving a file from one collection to another doesn't work?

#19 - 03/29/2018 04:08 PM - Peter Amstutz

I assume this logic means the default / "root" projectnode is the current user, but shouldn't that be initialized by whoever created the projectnode?

```
    if pn.uuid == "" {
        var resp User
        pn.err = fs.RequestAndDecode(&resp, "GET", "arvados/v1/users/current", nil, nil)
        if pn.err != nil {
            return
        }
        pn.uuid = resp.UUID
    }
```

#20 - 03/29/2018 04:22 PM - Peter Amstutz

```
    for {
        var resp CollectionList
        pn.err = fs.RequestAndDecode(&resp, "GET", "arvados/v1/collections", nil, params)
        if pn.err != nil {
            return
        }
        if len(resp.Items) == 0 {
            break
        }
        for _, i := range resp.Items {
            coll := i
            if coll.Name == "" {
                continue
            }
            pn.inode.Child(coll.Name, func(inode) (inode, error) {
                return deferredCollectionFS(fs, pn, coll), nil
            })
        }
        params.Filters = append(filters, Filter{"uuid", ">", resp.Items[len(resp.Items)-1].UUID})
    }
```

1. Could paging be generalized in a function that returned results on a channel instead of having to be manually coded everwhere?
2. It looks like it just appends filters, so after the first loop you'll have [uuid > 123], the next loop you'll have [uuid > 123, uuid > 456], on the next loop you'll have [uuid > 123, uuid > 456, uuid > 789] etc? This looks like another case of confusion where where append(N) may or may not modify N?
3. Based on recent query optimization adventures, I'm curious if there is a performance difference with sorting by uuid and using a uuid filter compared to using offset.

```
    filters = append(filters, Filter{"group_class", "=", "project"})
```

Similarly, this may include all the [uuid > 123, ...] filters

#21 - 03/29/2018 05:50 PM - Tom Clegg

Peter Amstutz wrote:

> - is there a sensible way to split the branch so that the go FUSE stuff isn't getting merged along with the webdav changes?

Yes, I think I can split off the lib/mount dirs and the line that adds the mount subcommand to arvados-client.

> - can we stop giving things confusing names? like crunchrunner vs crunch-run and now arvadosclient vs arvados-client

arvados-client is the future replacement for "arv". When we get around to packaging things nicely we'll want an "arvados-client" package that has all the Apache-licensed stuff and none of the GPL-licensed stuff, and a shim or other mechanism so "arvados foo" runs Apache commands everywhere and GPL commands where available. arvadosclient was a bad name and we're fixing that by deleting it.

Given the different contexts (go library vs. command line tool) I think it's not necessarily bad to have two different things called "arvados".

But we digress...

> I don't know if this was on your master plan, but apparently cgofuse supports WinFSP, which means a future Go FUSE driver could run natively on Windows, as an alternative to our current plan to use Windows' WebDAV support?

Yes, that's the biggest reason for trying cgofuse before bazil. I've built windows and osx binaries but I haven't gotten as far as finding systems to test them on. The notes for the fuse stuff are back at #12308.

Peter Amstutz wrote:

> Apologies if some of these comments arn't directly related to the branch, the diff is very large and includes refactoring of existing code (which appears as additions in the patch) so I'm wandering around a bit trying to get a handle on it.

Yeah, lots happened. I tried to make a sensible git log, but reading that might be just as tedious. (Is there a git diff mode that tries harder to show "moved" vs. deleted + added...?)

> [...]
> This comment need more information.
>
> - "reading and writing files in a FileSystem." but presumably a given instance only refers to one file

Right, that should probably say "... a file in a FileSystem".

> - Since it has Readdir, It can also represent a directory?
> - it's more like a filehandle, and the underlying file data and metadata is represented by an inode?

Yes to both -- just like os.File.

> [...]
> This is a confusing, is this talking about separate file system mounts (from the kernel POV) or Filesystem objects within the Go program? Does this mean moving a file from one collection to another doesn't work?

Different FileSystem objects. (I should camelcase that.)

Yes, it means renaming files across collections doesn't work.

Peter Amstutz wrote:

> I assume this logic means the default / "root" projectnode is the current user, but shouldn't that be initialized by whoever created the projectnode?

That works too, but if you ask for a projectnode with uuid=="" then it gives you "home" behavior instead of an error. Aside from being convenient it means you can mount /home in your filesystem but avoid using/waiting on an API call unless/until it's actually used.

> 1. Could paging be generalized in a function that returned results on a channel instead of having to be manually coded everwhere?

Yes, a filepath.Walk-like thing would be a good addition to the client library. (I think callers who want a channel should just supply a walkfn that sends to a channel, though -- foisting concurrency patterns on callers tends to be annoying.)

We can try a private one for now, and make it public when we decide it's an API worth supporting.

> 1. It looks like it just appends filters, so after the first loop you'll have [uuid > 123], the next loop you'll have [uuid > 123, uuid > 456], on the next loop you'll have [uuid > 123, uuid > 456, uuid > 789] etc? This looks like another case of confusion where where append(N) may or may not modify N?

Ever-growing filters list would happen if it were filters=append(filters,...) but this is params.Filters=append(filters,...) -- the invariant is "params.Filters is filters plus the current paging condition."

append() might modify the filters slice's backing array, but that's OK because it's only ever accessed by this one goroutine.

> 1. Based on recent query optimization adventures, I'm curious if there is a performance difference with sorting by uuid and using a uuid filter compared to using offset.

Possibly. This is more predictable.

> Similarly, this may include all the [uuid > 123, ...] filters

(addressed above -- filters never includes a paging condition.)

#22 - 03/29/2018 05:56 PM - Peter Amstutz

```
var (
    ErrReadOnlyFile      = errors.New("read-only file")
    ErrNegativeOffset    = errors.New("cannot seek to negative offset")
    ErrFileExists        = errors.New("file exists")
    ErrInvalidOperation  = errors.New("invalid operation")
    ErrInvalidArgument   = errors.New("invalid argument")
    ErrDirectoryNotEmpty = errors.New("directory not empty")
    ErrWriteOnlyMode     = errors.New("file is O_WRONLY")
    ErrSyncNotSupported  = errors.New("O_SYNC flag is not supported")
    ErrIsDirectory       = errors.New("cannot rename file to overwrite existing directory")
    ErrNotADirectory     = errors.New("not a directory")
    ErrPermission        = os.ErrPermission
)
```

Could these be returned embedded in an os.PathError?

#23 - 03/29/2018 06:52 PM - Peter Amstutz

Using gnome file browser, I tried opening it using dav://

1. If you don't try to mount /users you get an error, maybe the root should return a listing with just a "users" folder?
2. Users displayed the users just fine
3. I tried to navigate to one of the users. It waited for a while (maybe 20-30 seconds?) and then returned an error. Here's the log:

```
2018-03-29_18:48:12.97985 2018/03/29 18:48:12 "172.17.0.1:54552" 207 "Multi-Status" 594 "PROPFIND" "172.17.0.2
:9002" "/users/a" ""
```

```
2018-03-29_18:48:40.18259 2018/03/29 18:48:40 http: multiple response.WriteHeader calls
2018-03-29_18:48:40.18261 2018/03/29 18:48:40 error from webdav handler: "file does not exist"
2018-03-29_18:48:40.18261 2018/03/29 18:48:40 "172.17.0.1:54552" 500 "Internal Server Error" 146233 "PROPFIND"
 "172.17.0.2:9002" "/users/a/" ""
```

#24 - 03/29/2018 07:47 PM - Peter Amstutz

Cadaver seems to be failing the same way:

```
dav:/a/? open http://172.17.0.2:9002/users/a
Authentication required for collections on server `172.17.0.2':
Username: none
Password: 
dav:/users/a/> ls
Listing collection `/users/a/':
 failed:
XML parse error at line 1: Extra content at the end of the document
```

```
2018-03-29_19:44:07.82747 2018/03/29 19:44:07 "172.17.0.1:38050" 0 "" 0 "OPTIONS" "172.17.0.2:9002" "/users/a/
" ""
2018-03-29_19:44:07.90527 2018/03/29 19:44:07 "172.17.0.1:38050" 207 "Multi-Status" 591 "PROPFIND" "172.17.0.2
:9002" "/users/a/" ""
2018-03-29_19:44:31.61679 2018/03/29 19:44:31 http: multiple response.WriteHeader calls
2018-03-29_19:44:31.61682 2018/03/29 19:44:31 error from webdav handler: "file does not exist"
2018-03-29_19:44:31.61682 2018/03/29 19:44:31 "172.17.0.1:38050" 500 "Internal Server Error" 219216 "PROPFIND"
 "172.17.0.2:9002" "/users/a/" ""
```

#25 - 03/29/2018 07:47 PM - Peter Amstutz

This is just an arvbox test user but it probably has thousands of collections.

#26 - 03/29/2018 08:01 PM - Peter Amstutz

There's a length comment here but it doesn't say whether the Child() method takes a path or only names of direct children?

```
// Child() performs lookups and updates of named child nodes.
```

#27 - 03/29/2018 08:15 PM - Peter Amstutz

Shouldn't it filter for "." and ".." on collection names as well?

```
func (pn *projectnode) load() {
...
            if coll.Name == "" {
                continue
            }
...
            if group.Name == "" || group.Name == "." || group.Name == ".." {
                continue
            }
}
```

#28 - 03/29/2018 08:23 PM - Peter Amstutz

Aha, maybe the problem is that it is returning HTML and not DAV XML?

```
$ curl -H "Authorization: Bearer 1pydv91bffip0sv10cq1k5f7qvndxou745h4snwljme3lk73v1" http://172.17.0.2:9002/us
ers/a
<!DOCTYPE HTML>
<HTML><HEAD>
...
```

#29 - 03/29/2018 08:52 PM - Peter Amstutz

Oh, maybe this is the problem:

```
  <LI><A href="12213-keepref-wf.cwl/">12213-keepref-wf.cwl/</A></LI>
```

When I'm at "http://172.17.0.2:9002/users/a" and click on that link, it tries to go to "http://172.17.0.2:9002/users/12213-keepref-wf.cwl" and not "http://172.17.0.2:9002/users/a/12213-keepref-wf.cwl"

#30 - 03/29/2018 09:11 PM - Peter Amstutz

I navigated to a subfolder of home via Workbench and then via WebDAV. The workbench page loaded in 5 seconds, the WebDAV page took more like 30 seconds.

I think the problem is that it is loading the entire listing for home to determine if a single collection/subfolder exists. If we are not caching the listing between requests (or really, even if we do) I think projectnode.Child() needs to be able to look single entries to get reasonable performance.

#31 - 04/06/2018 08:53 PM - Tom Clegg

Peter Amstutz wrote:

> Using gnome file browser, I tried opening it using dav://
>
> 1. If you don't try to mount /users you get an error, maybe the root should return a listing with just a "users" folder?

Seems reasonable. Now "/" lists "users" and "by_id" instead of error 404.

> 1. I tried to navigate to one of the users. It waited for a while (maybe 20-30 seconds?) and then returned an error. Here's the log:
>
> 2018-03-29_18:48:40.18261 2018/03/29 18:48:40 "172.17.0.1:54552" 500 "Internal Server Error" 146233 "PROPFIND" "172.17.0.2:9002" "/users/a/" ""
>
> Cadaver seems to be failing the same way:

I haven't been able to reproduce this, but I wonder whether it's related to the "trailing slash on dir" bug (see next point):

> When I'm at "http://172.17.0.2:9002/users/a" and click on that link, it tries to go to "http://172.17.0.2:9002/users/12213-keepref-wf.cwl" and not "http://172.17.0.2:9002/users/a/12213-keepref-wf.cwl"

Indeed, I missed the "redirect /foo to /foo/ if /foo is a directory" thing in serveSiteFS. I've added that, along with tests, including tests for the same behavior in the non-serveSiteFS cases.

> Aha, maybe the problem is that it is returning HTML and not DAV XML?

I think this is OK. GET should return HTML (or, in that specific case, a redirect). WebDAV uses PROPFIND, which should return XML.

> There's a length comment here but it doesn't say whether the Child() method takes a path or only names of direct children?

It does not. In "Child() performs lookups and updates of named child nodes", "child" really just means child, not grandchild or anything else. Added a sentence to clarify (name is not . or .. and does not contain /).

> Shouldn't it filter for "." and ".." on collection names as well?

Yes, added.

> I navigated to a subfolder of home via Workbench and then via WebDAV. The workbench page loaded in 5 seconds, the WebDAV page took more like 30 seconds.
>
> I think the problem is that it is loading the entire listing for home to determine if a single collection/subfolder exists. If we are not caching the listing between requests (or really, even if we do) I think projectnode.Child() needs to be able to look single entries to get reasonable performance.

Yes, this would be a good optimization. Do you think it should block a merge?

13111-webdav-projects @ 61a5f1cab58f4652a9f9bdd4f6bc26c887eae75b

#32 - 04/09/2018 07:27 PM - Peter Amstutz

Tom Clegg wrote:

I haven't gotten back to looking at the branch yet, just responding to this one:

> > I navigated to a subfolder of home via Workbench and then via WebDAV. The workbench page loaded in 5 seconds, the WebDAV page took more like 30 seconds.
> >
> > I think the problem is that it is loading the entire listing for home to determine if a single collection/subfolder exists. If we are not caching the listing between requests (or really, even if we do) I think projectnode.Child() needs to be able to look single entries to get reasonable performance.
>
> Yes, this would be a good optimization. Do you think it should block a merge?

I think it is effectively unusable without it, so yes. (When we discussed it before I think I said one API call per path segment was an acceptable tradeoff for a first pass, but this is much worse).

> 13111-webdav-projects @ 61a5f1cab58f4652a9f9bdd4f6bc26c887eae75b

#33 - 04/10/2018 05:25 PM - Peter Amstutz

Tom Clegg wrote:

> 13111-webdav-projects @ 61a5f1cab58f4652a9f9bdd4f6bc26c887eae75b

This doesn't seem to have been pushed.

#34 - 04/10/2018 06:52 PM - Peter Amstutz

Getting the root over webdav works. The "/users" path works, but the "/by_id/" path doesn't:

```
2018-04-10_18:44:42.74687 2018/04/10 18:44:42 "172.17.0.1:54770" 404 "Not Found" 0 "PROPFIND" "172.17.0.2:9002
" "/by_id/" ""
```

Either it should do something reasonable, or it shouldn't return it.

When I try to get a listing of a user's home it still fails:

```
2018-04-10_18:45:00.78175 2018/04/10 18:45:00 "172.17.0.1:54770" 207 "Multi-Status" 591 "PROPFIND" "172.17.0.2
:9002" "/users/a/" ""
2018-04-10_18:45:58.53421 2018/04/10 18:45:58 http: multiple response.WriteHeader calls
2018-04-10_18:45:58.53423 2018/04/10 18:45:58 error from webdav handler: "file does not exist"
2018-04-10_18:45:58.53424 2018/04/10 18:45:58 "172.17.0.1:54770" 500 "Internal Server Error" 360331 "PROPFIND"
 "172.17.0.2:9002" "/users/a/" ""
```

```
dav:/users/> cd a
dav:/users/a/> ls
Listing collection `/users/a/': failed:
XML parse error at line 1: Extra content at the end of the document
```

I'll try to get more information.

#35 - 04/10/2018 07:18 PM - Peter Amstutz

- It seems the text "Internal Server Error" is literally being added to the end of the XML response:

```
$ curl -X PROPFIND -H "Authorization: Bearer 1pydv91bffip0sv10cq1k5f7qvndxou745h4snwljme3lk73v1" http://172.17
.0.2:9002/users/a/
<?xml version="1.0" encoding="UTF-8"?><D:multistatus xmlns:D="DAV:">...
...</D:multistatus>Internal Server Error
```

- This section from doc.go is out of date:

```
    // Indexes
    //
    // Currently, keep-web does not generate HTML index listings, nor does
    // it serve a default file like "index.html" when a directory is
    // requested. These features are likely to be added in future
    // versions. Until then, keep-web responds with 404 if a directory
    // name (or any path ending with "/") is requested.
```

#36 - 04/10/2018 09:56 PM - Peter Amstutz

There's two problems here:

1. Finally figured out the mysterious "file does not exist" error is caused by a collection with the name "Docker image arvados/jobs:1.1.3.20180406152156", it is getting confused by the '/', from discussion we agreed the immediate fix is to filter out invalid filenames
2. There's the error reporting issue where an error results in it responding with status code 500, the PROPFIND XML response, and the text "Internal Server Error" tacked onto the end of the body, which causes a parse error on the client.

#37 - 04/11/2018 03:08 PM - Tom Clegg

- Target version changed from 2018-04-11 Sprint to 2018-04-25 Sprint

#38 - 04/12/2018 05:48 PM - Tom Clegg

13111-webdav-projects @ e6c9fc68cd6d4d281d81928729a2bd18007a96a4

- Directory listings don't include projects and collections with "/" in their names (they can't be opened/stated anyway)
- Apply attachment policy (XSS protection) to /users/ and /by_id/
- Serve /by_id/ directory (was listed, but not routed to sitefs)
- Fix double-WriteHeader bugs
- Update docs

#39 - 04/13/2018 07:41 PM - Tom Clegg

13111-webdav-projects @ 73b36db80121301aaf3c3a38e313e1cbc62b3fdd

- avoid retrieving multiple [pages of] items until a caller asks for a dir listing

Caching from one request to the next will be necessary to make this usable as a mounted filesystem. I think the way to do this is to cache one sitefs per token, each protected with an RWMutex and discarded on write errors.

Meanwhile, this gets us to O(N) instead of O(N×M) on paths N levels deep with M children on each level.

Fixes for bugs encountered while testing:

- unhandled errors in cgofuse driver
- keepclient too sensitive about response transfer encoding (didn't work with 9tee4 cluster)

#40 - 04/13/2018 08:48 PM - Tom Clegg

Git shenanigans to remove "arvados-client mount" from this branch:

```
(13111-webdav-projects)$ git filter-branch -f --tree-filter 'git diff master...${GIT_COMMIT} build cmd | patch
 --silent -f -R -p1 && rm -rf lib/mount services/mount cmd/arvados-client/Makefile' --msg-filter 'sed -e "s/12
308:/13111:/"' --prune-empty -- master..HEAD
```

↓

13111-webdav-projects @ fd95e7a933b1534b76e4820838e278595b4e1220

...then merged master.

13111-webdav-projects @ 98911cfe4792b20798858cefb353c451460e1a80

https://ci.curoverse.com/job/developer-run-tests/688/

#41 - 04/20/2018 03:50 PM - Peter Amstutz

- So far so good. I can get a listing of "home" now (it takes a long time, but it works). I can also enter folders under "home" without waiting a long time.

- "by_id" works for collections. I guess we decided not to make it work for projects?

- Who's responsible for transforming "+" into a space? I thought we decided not to do things like that? Shouldn't this be /My%20Project/My%20Collection/? I don't see any tests using "+" for space?

  // If the collection is named "My Collection" and located in a project
  // called "My Project" which is in the home project of a user with
  // username is "bob", the following read-only URL is also available
  // when authenticating as bob:
  //
  //   http://collections.example.com/users/bob/My+Project/My+Collection/foo/bar.txt

#42 - 04/20/2018 03:59 PM - Peter Amstutz

I got failures running tests in arvbox. Running tests on jenkins here: https://ci.curoverse.com/job/developer-run-tests/690/

#43 - 04/20/2018 05:51 PM - Peter Amstutz

In arvbox I get a bunch of test failures like this:

```
cadaver_test.go:271:
    c.Check(stdout, check.Matches, `(?ms).*collection is empty.*`)
... value string = "" +
...     "Authentication required for collections on server `127.0.0.1':\n" +
...     "Username: ls\n" +
...     "Authentication aborted!\n" +
...     "Could not open collection:\n" +
...     "Could not authenticate to server: rejected Basic challenge\n" +
...     "dav:/by_id/? \n"
... regex string = "(?ms).*collection is empty.*"
```

#44 - 04/20/2018 06:59 PM - Tom Clegg

Peter Amstutz wrote:

> - "by_id" works for collections. I guess we decided not to make it work for projects?

Sort of. More like by_id is out of scope. Having anything at all there was an accidental bonus. Seems like we might as well leave it in, even if arv-mount gained projects while this was being reviewed and that makes them temporarily asymmetric?

> // http://collections.example.com/users/bob/My+Project/My+Collection/foo/bar.txt

Oops, + only works in form-encoding, not paths. Changed to MyProject etc.

> In arvbox I get a bunch of test failures like this:

Sounds like cadaver is not reading the $HOME/.netrc file set up by the test, but I don't know why it wouldn't.

#45 - 04/20/2018 07:58 PM - Peter Amstutz

Tom Clegg wrote:

> Peter Amstutz wrote:
>
> > - "by_id" works for collections. I guess we decided not to make it work for projects?
>
> Sort of. More like by_id is out of scope. Having anything at all there was an accidental bonus. Seems like we might as well leave it in, even if arv-mount gained projects while this was being reviewed and that makes them temporarily asymmetric?

That's fine.

> > // http://collections.example.com/users/bob/My+Project/My+Collection/foo/bar.txt
>
> Oops, + only works in form-encoding, not paths. Changed to MyProject etc.

Aha, good.

> > In arvbox I get a bunch of test failures like this:
>
> Sounds like cadaver is not reading the $HOME/.netrc file set up by the test, but I don't know why it wouldn't.

Thanks, that gives me something to investigate.

Looks good to me!

#46 - 04/23/2018 01:15 PM - Tom Clegg

13111-projects-by-id @ a93f2efa516c72475dd6f13872c0698c4c499aa9

...adds /by_id/$project_uuid/$subproject_name/$collection_name/...

#47 - 04/23/2018 06:21 PM - Peter Amstutz

The tests pass in arvbox if I replace this:

```
        cmd.Env = append(os.Environ(), "HOME="+tempdir)
```

with this:

```
        for _, r := range os.Environ() {
            if r[0:5] == "HOME=" {
                cmd.Env = append(cmd.Env, "HOME="+tempdir)
            } else {
                cmd.Env = append(cmd.Env, r)
            }
        }
```

The reasoning being that the 1st case results in $HOME being declared twice and it is ambiguous which one should be used.

#48 - 04/23/2018 06:38 PM - Peter Amstutz

LGTM if you add testing fix to runCadaver():

```
        for _, r := range os.Environ() {
            if r[0:5] != "HOME=" {
                cmd.Env = append(cmd.Env, r)
            }
        }
        cmd.Env = append(cmd.Env, "HOME="+tempdir)
```

#49 - 04/24/2018 03:28 AM - Tom Clegg

Peter Amstutz wrote:

> LGTM if you add testing fix to runCadaver():

Aha. Updated arvbox (and package-build) to go1.9.5 instead.

"The os/exec package now prevents child processes from being created with any duplicate environment variables. If Cmd.Env contains duplicate environment keys, only the last value in the slice for each duplicate key is used." -- https://golang.org/doc/go1.9

13111-projects-by-id @ 7f61e7cf4ae4661038092c6072bbeb437dc74146

#50 - 04/24/2018 02:02 PM - Peter Amstutz

Tom Clegg wrote:

> Peter Amstutz wrote:
>
>> LGTM if you add testing fix to runCadaver():
>
> Aha. Updated arvbox (and package-build) to go1.9.5 instead.
>
> "The os/exec package now prevents child processes from being created with any duplicate environment variables. If Cmd.Env contains duplicate environment keys, only the last value in the slice for each duplicate key is used." -- https://golang.org/doc/go1.9
>
> 13111-projects-by-id @ 7f61e7cf4ae4661038092c6072bbeb437dc74146

I'm glad that Go 1.9 is smarter about handing environment variables, but I don't think "require upgrade of Go" is an adequate solution. keep-web tests are currently failing on master due to the same bug, and it requires engaging ops to upgrade Go.
https://ci.curoverse.com/job/run-tests-remainder/1883/consoleFull

#51 - 04/25/2018 03:16 PM - Tom Clegg

- Target version changed from 2018-04-25 Sprint to 2018-05-09 Sprint

#52 - 04/27/2018 06:16 PM - Tom Clegg

It seems that gvfs-dav (like cadaver), while it supports http authentication, declines to send the configured credentials unless it gets a 401 response to its first request. However, if keep-web is configured with an anonymous token, it serves a root directory with "by_id" and "users" dirs (rather than a 401) to unauthenticated clients. The result is that gvfs-dav can't open an authenticated session with a keep-web that also serves anonymous clients.

Possible fix: at "sitefs" paths (where "*" is a non-empty name with no slashes, and "**" is a non-empty path with zero or more slashes):

- Respond 401 to /, /*, /users/, and /by_id/ if no token is supplied in the request
- Respond 401 instead of 404 to /users/** and /by_id/** if no token is supplied in the request and the first two path components of the requested path correspond to a missing directory. Examples: If no token is supplied in the request...
  - GET "/by_id/zzzzz-4zz18-abcdeabcdeabcde/foo" => 404 if "/by_id/zzzzz-4zz18-abcdeabcdeabcde" exists but .../foo does not
  - GET "/by_id/zzzzz-4zz18-abcdeabcdeabcde/foo" => 401 if "/by_id/zzzzz-4zz18-abcdeabcdeabcde" does not exist

  - GET "/users/foo/bar" => 404 if "/users/foo" exists but "/users/foo/bar" does not exist
  - GET "/users/foo/bar" => 401 if "/users/foo" does not exist

#53 - 04/27/2018 07:20 PM - Peter Amstutz

Brainstorming here:

- Respond 401 to "/" with instructions to the user to either provide their token or provide the username "anonymous" and any password
- Any other path: if no credentials provided, try the anonymous token

#54 - 04/27/2018 07:22 PM - Tom Clegg

Peter Amstutz wrote:

> - Any other path: if no credentials provided, try the anonymous token

I think with this it would still be impossible to mount "//download.example.com/users/foo", for example.

Accepting a special user/pass that means "just use anonymous, don't ask me to authenticate" could help avoid 401 cycles. But I wonder if this should just be a separate URL (another configurable special vhost?) so it can be copy-pasted.

#55 - 04/27/2018 07:35 PM - Peter Amstutz

> But I wonder if this should just be a separate URL (another configurable special vhost?) so it can be copy-pasted.

Yes, along those lines, how about a mount point called /public (maybe containing /public/by_id and /public/shared) which only exposes the stuff shared with anonymous, and everything else returns a 401 challenge if there are no credentials.

#56 - 04/27/2018 08:26 PM - Tom Clegg

I think the most reasonable short-term fix is to completely disable anonymous access to the sitefs routes: just respond 401 if no token is provided.

Before trying to support anonymous browsing from the root dir, we should probably think through stuff like robots.txt, what happens when an anonymous client reads a publicly-readable collection and then tries to write to it, etc.

#57 - 04/30/2018 03:39 PM - Tom Clegg

13111-no-anonymous-sitefs @ 3167926a35521efb58550ef0e26fb8c9e3a8450b

#58 - 05/01/2018 06:02 PM - Peter Amstutz

Tom Clegg wrote:

> 13111-no-anonymous-sitefs @ 3167926a35521efb58550ef0e26fb8c9e3a8450b

This LGTM.

#59 - 05/09/2018 01:23 PM - Tom Clegg

- Status changed from In Progress to Resolved

#60 - 07/23/2018 06:41 PM - Tom Morris

- Release set to 13