Arvados - Bug #7884

[Workbench] log viewer does not handle "redirect to keep-web" response

12/01/2015 11:06 PM - Tom Clegg

Status:	Resolved	**Start date:**	12/01/2015
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	Workbench	**Estimated time:**	0.00 hour
Target version:	2016-01-06 sprint		

Description

The Log pane of a finished job does an AJAX request with a URL like this:

```
Accept:*/*

https://workbench.4xphq.arvadosapi.com/collections/f5ab63b9f751ebceb3f58693a130ec62+83/4xphq-8i9sb-wub5wkrd85xaja0.log.txt
```

When keep-web is enabled, Workbench's response looks like this:

```
Content-Type:application/json; charset=utf-8

{"href":"https://download.4xphq.arvadosapi.com/c=f5ab63b9f751ebceb3f58693a130ec62-83/_/4xphq-8i9sb-wub5wkrd85xaja0.log.txt?api_token=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"}
```

The code in source:apps/workbench/app/views/jobs/_show_log.html.erb crashes at data.split() because data is an Object, not a string:

```
    $.ajax(logcollection_url, { headers: range_header }).
        done(function(data, status, jqxhr) {
            logViewer.filter();
            addToLogViewer(logViewer, data.split("\n"), taskState);
```

At least two things should be fixed:

- The $.ajax() call should force the response to be treated as plain text regardless of whether the response looks like JSON or has a Content-Type header.
- The Workbench download handler, when redirecting to keep-web, should bypass ApplicationController#redirect_to (in source:apps/workbench/app/controllers/application_controller.rb) and send a real HTTP redirect response.

Subtasks:

Task # 8049: keep-web respond to AJAX without redirect	**Resolved**
Task # 8047: Workbench accept text/plain	**Resolved**
Task # 8048: Workbench follow redirect to keep-web (considering CORS)	**Resolved**
Task # 8034: review 7884-ajax-log-redirect	**Resolved**

Related issues:

Related to Arvados - Feature #8064: [Keep-web] Support CORS requests with Aut...	**New**	**12/18/2015**

Associated revisions

Revision 2b699dec - 12/18/2015 08:40 PM - Tom Clegg

Merge branch '7884-ajax-log-redirect' closes #7884

History

#1 - 12/02/2015 10:34 PM - Brett Smith

- Target version set to Arvados Future Sprints

#2 - 12/16/2015 08:11 PM - Brett Smith

- Target version changed from Arvados Future Sprints to 2016-01-06 sprint

#3 - 12/16/2015 08:29 PM - Tom Clegg

- Assigned To set to Tom Clegg

#4 - 12/17/2015 07:24 AM - Tom Clegg

7884-ajax-log-redirect @ 384a517, tests at https://ci.curoverse.com/job/developer-test-job/73/

#5 - 12/17/2015 07:09 PM - Tom Clegg

- Status changed from New to In Progress

#6 - 12/18/2015 05:13 PM - Peter Amstutz

Reviewing 7884-ajax-log-redirect @ 384a517

I think I finally understand more or less what's going on. Comments:

I would prefer to avoid using POST in this way, since it breaks HTTP semantics.

Since the current solution already modifies keep-web, it would be straightforward to add support for the OPTIONS method and then do a proper GET request with an "Authorization" header.

Alternately, since the underlying problem doing a GET request with "?api_token=xxx" is that it responds with a redirect that sets the arvados_api_token cookie which gets rejected by the browser. Wouldn't it be better to do one of:

1. Use the /t=xxx/ syntax which doesn't set a cookie
2. Add &no_cookie to the query to suppress the cookie redirect

In both cases, since it's an AJAX request it's not going to show up in browsing history.

#7 - 12/18/2015 07:23 PM - Tom Clegg

Peter Amstutz wrote:

> I would prefer to avoid using POST in this way, since it breaks HTTP semantics.

You mean by making the response non-cacheable when by all rights it should be cacheable?

> Since the current solution already modifies keep-web, it would be straightforward to add support for the OPTIONS method and then do a proper GET request with an "Authorization" header.

Yes, as discussed on IRC, that's the real solution. But I'd like to merge this in the meantime, to unblock the work that's waiting for a bugfix.

> Alternately, since the underlying problem doing a GET request with "?api_token=xxx" is that it responds with a redirect that sets the arvados_api_token cookie which gets rejected by the browser. Wouldn't it be better to do one of:
>
> 1. Use the /t=xxx/ syntax which doesn't set a cookie
> 2. Add &no_cookie to the query to suppress the cookie redirect

I did consider a no_cookie param, as an explicit way to ask for this; OTOH, the Origin header is automatically added by the browser in exactly the situations where it will reject the redirect-with-cookie, so it seems like we would still want to do the "no_cookie" thing if Origin is set and no_cookie isn't.

> In both cases, since it's an AJAX request it's not going to show up in browsing history.

True, but I'd say it's also worth keeping tokens out of log files whenever we can.

#8 - 12/18/2015 07:50 PM - Peter Amstutz

Tom Clegg wrote:

> Yes, as discussed on IRC, that's the real solution. But I'd like to merge this in the meantime, to unblock the work that's waiting for a bugfix.

Do we have a ticket for that?

> True, but I'd say it's also worth keeping tokens out of log files whenever we can.

Okay, so the specific reasoning for using POST in this case instead of one of the alternatives is to avoid including the token in the URL so it doesn't get logged. That seems like a reasonable trade off.

#9 - 12/18/2015 08:32 PM - Peter Amstutz

This comment is misleading:

```
    if r.Header.Get("Origin") != "" {
        // Allow simple cross-origin requests, without
        // credentials.
        w.Header().Set("Access-Control-Allow-Origin", "*")
    }
```

What this is actually saying is this will permit CORS requests that don't use a custom Authorization header, but the credentials are supplied by some other mechanism.

#10 - 12/18/2015 08:34 PM - Tom Clegg

> Do we have a ticket for that?

I was thinking of just doing it here, but yes, probably better to move it to an issue and prioritize separately. Added #8064

#11 - 12/18/2015 08:34 PM - Tom Clegg

Peter Amstutz wrote:

> This comment is misleading:
>
> [...]
>
> What this is actually saying is this will permit CORS requests that don't use a custom Authorization header, but the credentials are supplied by some other mechanism.

Ah, yes. I meant "credentials" in the CORS sense, not the general sense. I'll clarify.

#12 - 12/18/2015 08:43 PM - Peter Amstutz

0.5 story points to write it and 1.5 story points to review it.

Looks good to me. Please merge.

#13 - 12/18/2015 08:45 PM - Tom Clegg

- Status changed from In Progress to Resolved

- % Done changed from 75 to 100

Applied in changeset arvados|commit:2b699dec710d1f0719ec0471bc711a467ac34a95.