

Arvados - Feature #4024

[Workbench] PipelineInstances#index should scroll down to show older pipelines, and should have a search box (similar to the "pipelines" tab on the "show project" page)

09/29/2014 04:03 PM - Tom Clegg

Status:	Resolved	Start date:	10/28/2014
Priority:	Normal	Due date:	
Assigned To:	Radhika Chippada	% Done:	100%
Category:	Workbench	Estimated time:	0.00 hour
Target version:	2014-11-19 sprint		
Description			
Subtasks:			
Task # 4331: Add infinite scrolling to pipeline_instances page.			Resolved
Task # 4463: Add search filter to pipeline_instances page			Resolved
Task # 4386: Review branch: 4024-pipeline-instances-scroll			Resolved
Task # 4498: Fix double-load after fwd/back			Resolved
Task # 4466: Scrolling when search filter is used			Resolved
Related issues:			
Related to Arvados - Bug #4523: [Workbench] Search dialog giving error when s...		Resolved	12/08/2014
Related to Arvados - Bug #4476: [Workbench] Search dialog infinite scrolling ...		Resolved	12/11/2014

Associated revisions

Revision ff7b22c7 - 11/17/2014 05:36 PM - Radhika Chippada

closes #4024

Merge branch '4024-pipeline-instances-scroll'

History

#1 - 10/08/2014 05:51 PM - Ward Vandewege

- Target version changed from Arvados Future Sprints to 2014-10-29 sprint

#2 - 10/08/2014 07:48 PM - Radhika Chippada

- Assigned To set to Radhika Chippada

#3 - 10/28/2014 01:28 PM - Radhika Chippada

- Status changed from New to In Progress

#4 - 10/29/2014 06:09 PM - Tom Clegg

- Target version changed from 2014-10-29 sprint to 2014-11-19 sprint

#5 - 10/31/2014 08:54 PM - Radhika Chippada

How do we want to prevent page refresh on the dashboard refreshing the recently finished pipelines panel when the user has scrolled down and viewing "next" page(s)?

Do we want to "disable" page refresh to the panel in that scenario?

#6 - 11/05/2014 07:23 PM - Tom Clegg

- Category set to Workbench

At [50517f3](#)...

Rather than adding a new route we could use the @/?tab-pane=finished_pipelines (and name the partial _show_finished_pipelines.html).

Was there a reason for changing the "finished pipelines" content from <div class="row"> to a <table><tr>?

The refresh behavior is pretty awkward here: the entire content area of the dashboard refreshes itself, which causes the "finished" pane to be empty,

and *then* the loading of the finished pane begins. This can be addressed *partially* by the more recent reload pattern which was introduced for the dashboard(?) and is being further refined in [#4084](#): we can update all of the dashboard panes independently, and remove the auto-update from the container that encloses all of the panes.

This still leaves us with a conflict between infinite scroll and dynamic refresh: we have no way to insert/update rows automatically without clearing the entire result set and starting over from page 1. This will be slow, and even worse, it will cause the browser to scroll automatically to the top every ~15 seconds, which will be even more annoying than the auto-scroll-to-bottom log viewer bug! I think the best solution is to use a client side framework like Angular, but perhaps there is also a more expedient solution.

First thought: Say we provided (some) content updates with a flag instructing the client-side reloader to "find objects with selector X, and use them as replacements for existing elements with the same ID". Any new matching elements could be inserted into the DOM the appropriate order. Dealing correctly with updated content on page 4 would probably require the client to tell the server a range (or a list of UUIDS?) of objects it should render.

#7 - 11/07/2014 02:30 PM - Radhika Chippada

Discussed this feature with Tom some more. We agreed that instead of adding infinite scrolling to the dashboard panels, we should add infinite scrolling to individual pages such as pipeline_instances page.

#8 - 11/07/2014 10:31 PM - Tom Clegg

At [7b4fc9f...](#)

The data-infinite-scroller attribute should be a jQuery selector string resolving to the element containing the rows (in this case, the tbody itself) so perhaps:

- ```
<tbody data-infinite-scroller="#recent-pipeline-instances" id="recent-pipeline-instances"
```

The search box target could be a bit more specific, to avoid matching (or even searching) any tables that happen to appear inside the rows themselves:

- ```
data-filterable-target="table.arv-recent-pipeline-instances > tbody"
```

- (Or even use the id of the tbody itself, if you add one as above)

Search/filter shouldn't require a remote form (I don't think we want Rails to do its remote:true ajax stuff here -- as is, the button generates a useless request, the result of which gets ignored somehow). Can it just be an input tag, like the search box in views/application/_choose.html.erb?

The params[:partial] handling in ApplicationController looks like it could fit inside the existing f.json{} instead of mirroring the logic. (As it stands, non-json with params[:partial] will render nothing at all, which doesn't seem like an important feature to add/preserve.)

I think we should get rid of the "pretend there is no next page if any filters were passed" stuff, and concentrate instead on making next_page_href (and everything else) propagate the filters to the next page correctly.

Isn't it redundant to use both the .html suffix and formats: [:html] here?

- ```
: "show_#{params[:partial]}.html", formats: [:html]
```

#### #9 - 11/08/2014 12:42 AM - Tom Clegg

- Subject changed from [Workbench] Dashboard "recent pipelines" panel should scroll down to show older pipelines, and should have a search box (similar to the "pipelines" tab on the "show project" page) to [Workbench] PipelineInstances#index should scroll down to show older pipelines, and should have a search box (similar to the "pipelines" tab on the "show project" page)

#### #10 - 11/08/2014 04:12 AM - Radhika Chippada

Tom, addressed all your feedback. Also, search works with scrolling now for this page. Thanks.

#### #11 - 11/08/2014 02:34 PM - Radhika Chippada

Tom, question about apps/workbench/app/views/pipeline\_instances/index.html.erb :

Does the "compare" form defined at line 2 work? I selected a couple of instances and still did not see the button. I then removed the "disabled: true, style: 'display: none'", just for testing, and clicking on the button (after selecting 2 or 3) gives Not found error for the URL "https://localhost:3031/pipeline\_instances/compare?utf8=%E2%9C%93&commit=Compare+2+or+3+selected". Not sure if it worked before but was broken somewhere along the way or never worked. I don't think I found any tests for it. Let me know if you would like me to file a bug for this.

```
<%= form_tag({action: 'compare', controller: params[:controller], method: 'get'}, {method: 'get', id: 'compare', class: 'pull-right small-form-margin'}) do |f| %>
 <%= submit_tag 'Compare 2 or 3 selected', {class: 'btn btn-primary', disabled: true, style: 'display: none'}
```

```
%>

<% end rescue nil %>
```

## #12 - 11/12/2014 06:19 AM - Tom Clegg

Radhika Chippada wrote:

Does the "compare" form defined at line 2 work?

This seems to have been broken by the tab loading code: the handler wasn't getting attached because the checkboxes weren't present at initial page load time. I've pushed a fix, [6c7bfabc](#), to your 4024 branch. Unfortunately it interferes with the layout (the search box next to the "compare" button looks a bit odd).

## #13 - 11/12/2014 06:27 AM - Tom Clegg

Radhika Chippada wrote:

Tom, addressed all your feedback. Also, search works with scrolling now for this page. Thanks.

Looks good! Couple of notes:

I suspect this addition is no longer needed -- is that right?

```
• if params[:search].andand.length.andand > 0
+ @select ||= PipelineInstance.columns.map(&:name)
+ base_search = PipelineInstance.select(@select)
+ @objects = base_search.where(any: ['contains', params[:search]]).
+ uniq { |pi| pi.uuid }
+ end
+
```

I ran into a double-load bug which it would be nice to fix, especially since pipeline instances load so slowly:

1. View the pipeline instances page
2. Type a string in the search box
3. Click one of the pipelines to show the pipeline\_instances#show page
4. Hit the browser "back" button
5. The instances page loads the first page of results
6. As soon as the first page is loaded, it erases and reloads again.

I suspect this means the first "first page load" AJAX happens before the filter state is properly set up. I think it would be a good idea to replicate this problem as a failing test case. If the solution isn't apparent, I wouldn't mind poking at it.

## #14 - 11/16/2014 05:28 AM - Tom Clegg

Now at [5d87bca](#) after

- merging 4388-workbench-update to eliminate javascript errors that were breaking workbench tests
- merging master
- fixing some resulting breakage (related to [#3400](#)).
  - [06afd90](#) 3400: Do not fetch\_multiple\_pages in #index actions.
    - This caused infinite scroll to get all pages instead of one page at a time.
  - [83e73ed](#) 4024: @limit override must happen before find\_objects\_for\_index.
    - @limit=20 in pipeline\_instances#index was being ignored.
  - [5d87bca](#) 3400: Do not fetch API results just for the sake of looking up resource\_class.
    - This used to cause one unnecessary API call; with 3400 it got much worse, causing an unnecessary fetch\_multiple\_pages

## #15 - 11/16/2014 11:10 PM - Radhika Chippada

Tom,

- pipeline\_instances page with no search string, with a matching search string works. However, if I were to enter a search filter with no matches (basically, some random / junk string), I see "Oops, request failed" error page. It appears that the response from API server seems incomplete / incorrect in this case.

```
#<NoMethodError: undefined method `+' for nil:NilClass>
/home/radhika/arvados/apps/workbench/app/controllers/application_controller.rb:227:in `next_page_offset'
/home/radhika/arvados/apps/workbench/app/controllers/application_controller.rb:238:in `next_page_href'
/home/radhika/arvados/apps/workbench/app/controllers/application_controller.rb:174:in `block (2 levels) in render_index'
/home/radhika/arvados/apps/workbench/vendor/bundle/ruby/2.1.0/gems/actionpack-4.1.1/lib/action_controller/metal/mime_responds.rb:258:in `call'
/home/radhika/arvados/apps/workbench/vendor/bundle/ruby/2.1.0/gems/actionpack-4.1.1/lib/action_controller/metal/mime_responds.rb:258:in
```

`respond\_to`

/home/radhika/arvados/apps/workbench/app/controllers/application\_controller.rb:171:in `render\_index`

/home/radhika/arvados/apps/workbench/app/controllers/application\_controller.rb:214:in `index`

- I updated the integration test that is using "junk" search filter to assert that there is no error in page; this test fails for the time being due to this issue.

**#16 - 11/17/2014 05:40 PM - Radhika Chippada**

- *Status changed from In Progress to Resolved*

- *% Done changed from 80 to 100*

Applied in changeset arvados|commit:ff7b22c70cd77073d9bdbebac0bf03d43745ed0c.