

Arvados - Bug #4227

[Workbench] Pipeline elapsed time is misformatted

10/16/2014 05:29 PM - Bryan Cosca

Status:	Resolved	Start date:	10/21/2014
Priority:	Normal	Due date:	
Assigned To:	Radhika Chippada	% Done:	100%
Category:	Workbench	Estimated time:	0.00 hour
Target version:	2014-10-29 sprint		
Description			
In pipeline_instances/qr1hi-d1hrv-43y08obrh2vetuv, it says This pipeline started at 11:56 AM 10/16/2014. It failed after 4 minutes 52 seconds at 12:01 AM 10/16/2014.			
I expected it to say 12:01 "PM"			
Subtasks:			
Task # 4277: Review branch: 4227-date-display			Resolved

Associated revisions

Revision 6eaaae29 - 10/22/2014 06:57 PM - Radhika Chippada

closes #4227
Merge branch '4227-date-display'

Revision ea80735a - 10/23/2014 07:37 PM - Radhika Chippada

refs #4227 - rescue from any error in parsing pipeline start and finish time so that jenkins runs do not choke on these failures.

Revision 4265a54a - 10/23/2014 08:22 PM - Radhika Chippada

refs #4227 - update strptime format for parsing pipeline start and finish time.

Revision 1ed38089 - 10/23/2014 08:36 PM - Radhika Chippada

refs #4227
Merge branch '4227-test-fix'

History

#1 - 10/17/2014 05:26 PM - Brett Smith

- Subject changed from Time log is wrong to [Workbench] Pipeline elapsed time is misformatted
- Category set to Workbench

#2 - 10/21/2014 05:40 PM - Radhika Chippada

- Status changed from New to In Progress
- Assigned To set to Radhika Chippada
- Target version set to 2014-10-29 sprint

The issue is in dates.js, another one of the under-tested javascript areas. Fix issue and add tests.

#3 - 10/21/2014 07:14 PM - Ward Vandewege

- Story points set to 0.5

#4 - 10/22/2014 03:34 PM - Brett Smith

Reviewing [e102efba](#). The fix is good, but I'm surprised at how involved the test code is.

- I'm not sure I understand what value we get out of testing ten cases of this. It seems like we get the most value just by testing a simple case, a case that spans the AM/PM boundary, and a case that spans the midnight boundary. If there are other cases that have some unique property, then let's definitely include those. But ten seems like an arbitrary number, and that's a lot of overhead given that we're firing up a whole browser session to test datetime formatting.
 - I now limited to two tests, one with 0 run time and one with run time that spans between AM and PM

- You can use [DateTime::strptime](#) to parse these strings, rather than doing all the splitting and array indexing yourself.
 - Thanks for this tip. It is fantastic
- Rather than trying to parse the "elapsed time" string, let's use fixtures whose elapsed time we know ahead of time, and compare against that. That will save us a bunch of parsing and math that aren't directly relevant to what we're trying to test (the datetime formatting).
 - Adjusted this

BTW, there's also trailing whitespace on some of the lines in the current branch.

Thanks.

#5 - 10/22/2014 06:47 PM - Brett Smith

Thanks for following through on this. [99ad159](#) is good to merge, thanks.

#6 - 10/22/2014 07:00 PM - Radhika Chippada

- *Status changed from In Progress to Resolved*

Applied in changeset arvados|commit:6eaaae29a7af005e417673d79e0951122065e685.