

openQA Project - action #92746

coordination # 39719 (Blocked): [saga][epic] Detection of "known failures" for stable tests, easy test results review and easy tracking of known issues

coordination # 19720 (Blocked): [epic] Simplify investigation of job failures

Log viewer in openQA webUI with color parsing

2021-05-17 11:28 - okurz

Status:	Resolved	Start date:	2021-05-17
Priority:	Low	Due date:	
Assignee:	mkittler	% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			
Description			
Motivation			
<p>Log files like autoinst-log.txt are often crucial for reviewing test results. openQA does not yet have a "log viewer", it only provides a download and browsers can choose to show logs verbatim. Compare to gitlab in https://gitlab.suse.de/openqa/auto-review/-/jobs/345765/raw showing the raw log with control codes from both code that comes from us as well as gitlab internal code doing the same in comparison to a nice log viewer https://gitlab.suse.de/openqa/auto-review/-/jobs/345765 that shows color for both parts again, e.g. nice colorful autoinst-log.txt . We also have a separate video viewer since https://github.com/os-autoinst/openQA/pull/2605 so we can try a similar approach for logs now</p>			
Acceptance criteria			
<ul style="list-style-type: none">• AC1: autoinst-log.txt accessible over the openQA webUI show colors based on the control characters included in autoinst-log.txt• AC2: Raw unaltered logs can still be accessed over the webUI for download			
Suggestions			
<ul style="list-style-type: none">• Research how other programs do log parsing, e.g. gitlab• Read what #65271 does now, e.g. especially https://progress.opensuse.org/issues/65271#note-54 and https://hackweek.suse.com/20/projects/create-a-library-and-demo-application-to-parse-the-openqa-logs-to-beautify-them-and-provide-query-tools and https://github.com/jlausuch/SUSE_Hackweek_2021_openQA_logs_analyzer_proof_of_concept• Implement a log viewer next to a download link, same as for videos, where we have "Video (download)", see https://github.com/os-autoinst/openQA/blob/master/templates/webapi/test/downloads.html.ep#L8			
Related issues:			
Related to openQA Project - action #94060: unstable test t/ui/18-tests-details.t		Resolved	2021-06-15 2021-07-01

History

#1 - 2021-05-17 11:34 - okurz

- Description updated

#2 - 2021-05-17 13:24 - okurz

- Target version changed from Ready to future

#3 - 2021-05-18 09:17 - tinita

There is a module for turning control characters into HTML, I've used it in the past: <https://metacpan.org/pod/HTML::FromANSI::Tiny>

#4 - 2021-05-18 12:09 - okurz

- Status changed from Workable to Feedback

- Assignee set to okurz

ok, cool idea. Unfortunately both <https://software.opensuse.org/package/perl-Parse-ANSIColor-Tiny> and https://software.opensuse.org/package/perl-HTML-FromANSI-Tiny?search_term=perl-HTML-FromANSI-Tiny are not yet in Factory and hence also

not in Leap. Oh well, another dependency. Will create submissions to Factory first.

<https://build.opensuse.org/request/show/894044>
<https://build.opensuse.org/request/show/894045>

#5 - 2021-05-18 12:27 - mkittler

Intuitively I would have implemented this on the client side. (Why keep our servers busy for something one can likely do with ease in JavaScript? A JavaScript solution has likely faster loading times.)

In a private project I've already tried xterm.js. It is very performant but also more of a terminal than a log view and searching and resizing is a bit inconvenient. However, there are likely JavaScript libraries out there which would fit our use-case.

#6 - 2021-05-18 13:00 - okurz

Sure, we don't need to use server side perl solutions. I just wanted to submit the packages in case we want to use them nevertheless. I would also favor a server side solution

#7 - 2021-05-18 13:33 - tinita

Client side also makes sense in this case.

Regarding keeping our servers busy: I have good experiences with caching (in redis, memcached, etc.). For example we are doing a lot of SQL queries on some pages, where we actually wouldn't have to redo those queries on every request (because the data doesn't change or because it would be ok to show slightly outdated data).
but that's a different story.

#8 - 2021-05-18 14:12 - okurz

tinita wrote:

Regarding keeping our servers busy: I have good experiences with caching (in redis, memcached, etc.). For example we are doing a lot of SQL queries on some pages, where we actually wouldn't have to redo those queries on every request (because the data doesn't change or because it would be ok to show slightly outdated data).
but that's a different story.

<https://progress.opensuse.org/projects/openqav3/issues/new> is the place for "different story" :)

#9 - 2021-05-18 17:31 - okurz

- Status changed from Feedback to Workable
- Assignee deleted (okurz)
- Target version changed from future to Ready

Both SRs accepted. This server side approach can be taken as secondary if the JS path fails

#10 - 2021-06-02 14:41 - mkittler

- Assignee set to mkittler

#11 - 2021-06-02 14:48 - mkittler

Seems like there a lot of JS libraries out there: <https://www.npmjs.com/search?q=keywords%3Aansi&ranking=popularity>
Maybe I should go with the most popular option - which would be in some way to create my own library and add it to npm :-)

By the way, it looks like CodeMirror (which we would already be using) doesn't support ANSI formatting.

#12 - 2021-06-09 12:02 - mkittler

- Status changed from Workable to In Progress

PR: <https://github.com/os-autoinst/openQA/pull/3936>

#13 - 2021-06-11 10:13 - mkittler

- Status changed from In Progress to Feedback

#14 - 2021-06-14 08:26 - mkittler

- Status changed from Feedback to Resolved

Works on OSD

#15 - 2021-06-15 18:31 - okurz

- Related to action #94060: unstable test t/ui/18-tests-details.t added