

openQA Project - action #93246

coordination # 91646 (Blocked): [saga][epic] SUSE Maintenance QA workflows with fully automated testing, approval and release

coordination # 89062 (Blocked): [epic] Simplify review for SUSE QAM

List all unreviewed failed (or incomplete) jobs on /tests on request size:M

2021-05-28 21:59 - okurz

Status: Workable	Start date: 2021-05-28
Priority: Low	Due date:
Assignee:	% Done: 0%
Category: Feature requests	Estimated time: 0.00 hour
Target version: Ready	
Difficulty:	

Description

Motivation

As a test reviewer and member of one of many QA squads I want to find unreviewed openQA test failures on /tests to have access to a single page showing all failures needing review work.

Also see parent [#89062](#)

Acceptance criteria

- **AC1:** A query parameter "todo=1" on /tests can be specified to show only jobs needing review, same as we do on /tests/overview

Suggestions

/tests can already be configured like https://openqa.opensuse.org/tests/?resultfilter=Failed&resultfilter=Incomplete&resultfilter=timeout_exceeded , even over UI elements.

- Try t/ui/01-list.t and see if you can extend it
- Add query parameter to filter only jobs that need review, e.g. "todo=1", same as on /tests/overview . At best use the same logic to prevent duplication

Out of scope

- Don't care about the "running" or "scheduled" jobs on /tests as long as they at least show something useful when no manual query parameters are used or no filtering box selection has been made.
- Performance improvements on /tests, as the feature should only be activated on explicit request by users
- Don't care about the number of jobs that are displayed after all the filters apply. <https://openqa.opensuse.org/tests?match=toolkits&resultfilter=Failed> is one example showing that there can be enough "review candidates" depending on the specified test suite
- Don't care about changing the limit of jobs (e.g. more than 500)
- Don't care about filtering the job for other parameters like job group, scenario name, result, state, etc.

Related issues:

Copied to openQA Project - action #94937: Distinguish comment types on jobs o...	Resolved	2021-05-28
--	-----------------	-------------------

History

#1 - 2021-05-31 07:07 - VANASTASIADIS

- Target version set to Ready

#2 - 2021-05-31 09:59 - mkittler

This would make the "All tests" page much less "All tests" like, e.g. I assume tables for running and scheduled jobs would be hidden completely.

Due to performance problems the number of finished jobs is by default limited to 500. Most existing filters on that page are implemented client-side and hence only limit these 500 jobs further. How is the new filter supposed to work? Also only within the most recent 500 jobs or should it go deeper so we would really show up to 500 unreviewed jobs? Then it needed to be implemented server-side like the existing "Show only relevant jobs" filter

(but still needed to handle query parameters like the filter for result types). I'm wondering how fast it would be, though. Going down within the job history (while parsing job comments for ticket references using Perl regex) until 500 unreviewed jobs have been found might take a while.

#3 - 2021-06-01 10:34 - cdywan

mkittler wrote:

Due to performance problems the number of finished jobs is by default limited to 500. Most existing filters on that page are implemented client-side and hence only limit these 500 jobs further. How is the new filter supposed to work? Also only within the most recent 500 jobs or should it go deeper so we would really show up to 500 unreviewed jobs? Then it needed to be implemented server-side like the existing "Show only relevant jobs" filter (but still needed to handle query parameters like the filter for result types).

I read the ticket to mean the API will filter the results based on the query - or is this actually all done client-side now? Your comment sounds like we have a bit of both.

#4 - 2021-06-02 14:17 - mkittler

Your comment sounds like we have a bit of both.

That's also the case. Filtering by result type and the search box are applied on the client-side (and only reduce the 500 jobs returned by the server).

By the way, I'm wondering when this would be useful. Maybe it is just me but I'm only using the "All tests" page on the production instances for running and scheduled jobs (and I suppose these tables would be hidden completely or shown unaffected). The list of finished jobs is rather arbitrary and due to the limit it goes only down to a few hours in the past. (Of course on my personal openQA instance it is another story - but I double people do much review work on their personal instances.)

#5 - 2021-06-08 09:34 - okurz

- Description updated

mkittler wrote:

This would make the "All tests" page much less "All tests" like, e.g. I assume tables for running and scheduled jobs would be hidden completely.

For the user story as define I suggest to completely ignore the impact on "running/scheduled" tests. At this point if expert users deliberately use a query parameter, e.g. over a manual entry in the address bar or even over a checkbox, they know what they ask for. If it makes it easier for us then, yes, for this mode we could make the "running / scheduled" boxes completely invisible.

Due to performance problems the number of finished jobs is by default limited to 500. Most existing filters on that page are implemented client-side and hence only limit these 500 jobs further. How is the new filter supposed to work? Also only within the most recent 500 jobs or should it go deeper so we would really show up to 500 unreviewed jobs? Then it needed to be implemented server-side like the existing "Show only relevant jobs" filter (but still needed to handle query parameters like the filter for result types).

Yes, the initial server-side query needs to know this request.

I'm wondering how fast it would be, though. Going down within the job history (while parsing job comments for ticket references using Perl regex) until 500 unreviewed jobs have been found might take a while.

Sure it can be slower than the default of just the latest 500 jobs but definitely not slower than users needing to manually open /tests/overview of 10 products :)

By the way, I'm wondering when this would be useful. Maybe it is just me but I'm only using the "All tests" page on the production instances for running and scheduled jobs [...] The list of finished jobs is rather arbitrary and due to the limit it goes only down to a few hours in the past.

Correct. Because I also think that power users on big openQA instances like openqa.opensuse.org and openqa.suse.de use /tests/overview I want to make /tests more useful for new use cases which we want to cover. If you are not sure what we want to achieve or why we want to do this please see the "Motivation" in the current ticket as well as the parent epic and saga with related tickets. I can also provide more context over other communication channels if you prefer.

(Of course on my personal openQA instance it is another story - but I double people do much review work on their personal instances.)

Actually I am sure that users rely more on /tests on smaller instances where "the last 500" make more sense, e.g. are only triggered by single users. So we must not break this use case.

#6 - 2021-06-08 10:21 - mkittler

Sure it can be slower than the default of just the latest 500 jobs but definitely not slower than users needing to manually open /tests/overview of 10 products :)

But it wouldn't be useful if the request times out, e.g. <https://openqa.opensuse.org/tests/overview> shows lots of tests on o3 but adding the TODO parameter here (<https://openqa.opensuse.org/tests/overview?todo=1>) results in a gateway timeout. I'm just saying that searching for 500 unreviewed jobs might take long, e.g. in the extreme case when all jobs have actually already been reviewed the whole list of jobs within the database needed to be checked (in order to return zero results).

If you are not sure what we want to achieve or why we want to do this please see the "Motivation" ...

The motivation "I want to find unreviewed openQA test failures on /tests to have access to a single page showing all failures needing review work" really raises the question: Unreviewed openQA tests since when? I'm sure users don't want to bother with very old failed jobs which have no relevance anymore.

#7 - 2021-06-08 13:17 - okurz

mkittler wrote:

Sure it can be slower than the default of just the latest 500 jobs but definitely not slower than users needing to manually open /tests/overview of 10 products :)

But it wouldn't be useful if the request times out, e.g. <https://openqa.opensuse.org/tests/overview> shows lots of tests on o3 but adding the TODO parameter here (<https://openqa.opensuse.org/tests/overview?todo=1>) results in a gateway timeout. I'm just saying that searching for 500 unreviewed jobs might take long, e.g. in the extreme case when all jobs have actually already been reviewed the whole list of jobs within the database needed to be checked (in order to return zero results).

yes, you are right. The difference to "manually open /tests/overview of 10 products" is that *first* we look up the *latest* job within each scenario and only then check if they are reviewed. So this is what we should do here as well. This is what I meant in the suggestions stating "Add [...] todo=1 same as on /tests/overview . At best use the same logic to prevent duplication"

If you are not sure what we want to achieve or why we want to do this please see the "Motivation" ...

The motivation "I want to find unreviewed openQA test failures on /tests to have access to a single page showing all failures needing review work" really raises the question: Unreviewed openQA tests since when? I'm sure users don't want to bother with very old failed jobs which have no relevance anymore.

Correct. As stated above we should look for the latest in each test scenario. And of course this can then reveal that "only jobs in the unstable development job group are left as unreviewed". This is why I suggested the optional "job group include/exclude regex" which is an optional implementation so depending on feasibility the implementors can decide if this is out of scope for the current story or not.

#8 - 2021-06-10 15:28 - okurz

- Priority changed from High to Low

As confirmed by vpelcak in DM where he addressed me in <https://chat.suse.de/direct/2FFxjQXvCCPbj5kbFYvRdSgu8mNq2StdK?msg=qKsBM6NKNqghTkuEF> we can focus on "openqa-review" related tasks with the expectation that this can act as the "better overview over their failing testcases" so we reduce priority here

#9 - 2021-06-29 15:39 - tinita

okurz wrote:

yes, you are right. The difference to "manually open /tests/overview of 10 products" is that *first* we look up the *latest* job within each scenario and only then check if they are reviewed. So this is what we should do here as well. This is what I meant in the suggestions stating "Add [...] todo=1 same as on /tests/overview . At best use the same logic to prevent duplication"

The difference to /tests/overview is that the list of jobs to load is determined by the query parameters.

What would "*latest* job within each scenario" here mean for the /tests page?

#10 - 2021-06-29 17:11 - okurz

What would "latest job within each scenario" here mean for the /tests page?

Same as already when you open /tests, the latest within each scenario is resolved. This means that older jobs for the same or older builds are hidden. For example if someone retriggers tests then only the latest clone is shown. Actually I am not so sure about "older builds" but I don't ask for any changes in this regard.

#11 - 2021-06-29 22:51 - tinita

okurz wrote:

What would "latest job within each scenario" here mean for the /tests page?

Same as already when you open /tests, the latest within each scenario is resolved. This means that older jobs for the same or older builds are hidden. For example if someone retriggers tests then only the latest clone is shown. Actually I am not so sure about "older builds" but I don't ask for any changes in this regard.

Jobs for older builds are shown, as long as they are in the latest 500.

I don't know about others, but I still wouldn't know how to approach this, given that there is no trivial way to filter the "reviewed" property with SQL. Getting the latest 500 failed via SQL and then filtering label/bugref with perl would work, and already this would require to fetch more data than for normal /tests queries (currently only bug counts are fetched).

#12 - 2021-06-29 23:05 - tinita

tinita wrote:

What would "latest job within each scenario" here mean for the /tests page?

Let me rephrase: What do you mean with "scenario" on the /tests page? Running, scheduled and finished?

#13 - 2021-06-30 08:33 - okurz

- Subject changed from *List all unreviewed failed (or incomplete) jobs on /tests* to *List all unreviewed failed (or incomplete) jobs on /tests on request*
- Description updated

tinita wrote:

okurz wrote:

What would "latest job within each scenario" here mean for the /tests page?

Same as already when you open /tests, the latest within each scenario is resolved. This means that older jobs for the same or older builds are hidden. For example if someone retriggers tests then only the latest clone is shown. Actually I am not so sure about "older builds" but I don't ask for any changes in this regard.

Jobs for older builds are shown, as long as they are in the latest 500.

I don't know about others, but I still wouldn't know how to approach this, given that there is no trivial way to filter the "reviewed" property with SQL.

Do you think the suggestions I made in the description regarding the "todo"-flag which we already have on /tests/overview does not help here?

Getting the latest 500 failed via SQL and then filtering label/bugref with perl would work, and already this would require to fetch more data than for normal /tests queries (currently only bug counts are fetched).

Well, I don't see a problem with that as I suggested that a new initial query is done based on query parameters. So this is not about "filtering pre-loaded data client-side with javascript" at this point.

tinita wrote:

tinita wrote:

What would "latest job within each scenario" here mean for the /tests page?

Let me rephrase: What do you mean with "scenario" on the /tests page? Running, scheduled and finished?

I mean scenario as described on
<http://open.qa/docs/#concepts>

<distri>-<version>-<flavor>-<arch>-<test_suite>@<machine>, e.g. "openSUSE-Tumbleweed-DVD-x86_64-gnome@64bit".

#14 - 2021-06-30 09:43 - okurz

- Copied to action #94937: Distinguish comment types on jobs on /tests (maybe optional) size:S added

#15 - 2021-06-30 09:50 - okurz

- Subject changed from List all unreviewed failed (or incomplete) jobs on /tests on request to List all unreviewed failed (or incomplete) jobs on /tests on request size:M

- Description updated

estimated together

#16 - 2021-07-09 11:52 - okurz

- Target version changed from Ready to future

#17 - 2021-07-14 12:43 - okurz

- Subject changed from List all unreviewed failed (or incomplete) jobs on /tests on request size:M to List all unreviewed failed (or incomplete) jobs on /tests on request

- Description updated

- Status changed from Workable to New

[#94937](#) solved. We can now continue here. To be re-estimated after prerequisites changed. Updated description.

#18 - 2021-07-14 14:50 - mkittler

I'm still wondering how the scenario would come into play here. Determining all present scenarios (unique combinations of <distri>-<version>-<flavor>-<arch>-<test_suite>@<machine>) to determine their latest job seems pretty expensive to do (across all jobs).

Or do you expect to supply parameters for a specific scenario when the TODO parameter is used? (But then the all tests page would just be a worse structured version of the test result overview page.)

#19 - 2021-07-16 14:05 - okurz

- Target version changed from future to Ready

#20 - 2021-08-26 12:14 - okurz

- Description updated

#21 - 2021-08-26 12:19 - tinita

- Subject changed from List all unreviewed failed (or incomplete) jobs on /tests on request to List all unreviewed failed (or incomplete) jobs on /tests on request size:M

- Status changed from New to Workable