

openQA Project - action #69976

Show dependency graph for cloned jobs

2020-08-13 09:31 - mkittler

Status:	Workable	Start date:	2020-08-13
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			
Description			
user story			
<ul style="list-style-type: none">As a tester or openQA developer I, I often follow links to openQA jobs (e.g. from a ticket) and for further investigation it would be useful to see the dependency tree - even though the job has already been cloned in the meantime.As a new user of openQA, I would like to avoid getting the wrong impression that dependencies of a job could not be created or have been deleted although they are just not displayed.			
acceptance criteria			
<ul style="list-style-type: none">AC0: The "Dependencies" tab is shown for jobs which have been cloned/restarted.AC1: The tab shown as in AC0 contains the actual job one is looking at (and <i>not</i> a different job in the "cloning chain").			
notes			
<p>Here a few points for consideration. I wouldn't call these ACs because it requires some experimentation to find out what works best in practice and can be implemented with reasonable effort. I only added these points because I'm already aware of certain cases which can be tricky (and have been tricky in the past) and should be thought through when implementing this issue.</p> <ol style="list-style-type: none">The dependency graph has been disabled on purpose so far: Showing not only all children but also all of their clones lead to overwhelmingly big graphs which we should keep avoiding. So far we avoid the problem by showing only the latest jobs of the "cloning chain" within the graph but that means cloned jobs have no graph at all.It would likely be useful to show graphs of old jobs as well. Users frequently get confused about missing graphs and sometimes use older jobs as reference.<ol style="list-style-type: none">Only dependencies of the same "cloning level" should be shown within a graph to avoid the problem described in 1..A dependency cluster might have only been cloned partially. Consider the example of restarting only a child but not the parent. Then there is not "one" cloning level. To avoid the problem described in 1. the graph of the parent job in this example should likely only show the most recent child like it does right now. The graph of the original child jobs should contain only the original child job. The graph of the cloned child jobs should contain only the cloned child job. 2.1. The last two statements are obvious but so far the graph of the original child job would not show the original child job at all but only contain the cloned child job.When skipping the restart of a passed/softaild child, the child is so far nevertheless added as child of to the cloned parent and therefore ends up as part of the old and the new cluster. So not only a parent (as explained in 2.2) but also a child might be part of multiple cloning levels. The graph rendering should treat both cases sensibly. To be consistent with 2.2 we should only show the most recent parent in the graph of the skipped child.			
suggestions			
<ol style="list-style-type: none">Try to understand the code in <code>OpenQA::WebAPI::Controller::Test::dependencies</code> and related functions. Reading the code of the JavaScript function <code>renderDependencyGraph</code> might be helpful as well. It should not be necessary to touch any other code because this is only about <i>displaying</i> dependencies.Change the condition <code>!defined(\$job->clone_id) && \$job->has_dependencies</code> to <code>\$job->has_dependencies</code>. That's all what's needed to implement AC0.Create some relevant job clusters to test AC1 and the scenarios mentioned under notes. One could either start testing manually (e.g. by using manual SQL statements to add the required dependencies) or by extending some of the dependencies tests.Play around with the code mentioned in 1. to implement AC1 considering the notes.			
Related issues:			
Related to openQA Project - action #59969: Display job dependency tab not onl...		Blocked	2019-11-18
Related to openQA Project - action #69979: Advanced job restarting via the we...		New	2020-08-13

History

#1 - 2020-08-13 17:56 - okurz

- Status changed from Workable to New
- Priority changed from Normal to Low
- Target version set to future

I don't consider it workable yet, e.g. missing user story and specific suggestions. Try to put yourself into the shoes of a junior contributor that asks "ok, what should I look into first, e.g. what topic to research, what source code to read, what test to write, what experiment to run"

#2 - 2020-08-20 13:56 - mkittler

- Description updated
- Status changed from New to Workable

I added use-cases and suggestions. However, I would actually like to work on this ticket on myself - this graph stuff is quite fun with all the recursion going on.

#3 - 2020-08-20 15:18 - Julie_CAO

Hi Marius,

Happy to see progress for this ticket. Each improvement from openQA side about the directly chained tests will benefit us. We are even in discussion if we should postpone our test reconstruction until directly chain gets more mature.

Here are the test chains we created FYI: <https://openqa.nue.suse.com/tests/overview?distri=sle&version=15-SP2&build=207.1&groupid=213>

Some questions about this ticket:

What does 'clone' here mean? If I restart tests in a directly chain on web UI, are these restarted tests called cloned jobs here? or those jobs which are restarted via openqa-clone-job?

If we only rerun part of tests(a sub tree of the original test chain tree), will the 'cloning tree' graph be the sub tree only? Can we still view or operate the original tree after implementing this ticket?

The graph is not just a graph, it is the actual chain, we can rerun jobs with the graph, do I get it correctly?

#4 - 2020-08-25 14:30 - mkittler

What does 'clone' here mean? If I restart tests in a directly chain on web UI, are these restarted tests called cloned jobs here?

Yes. Jobs which have been restarted via the web UI or direct use of the REST-API's restart and duplicate routes are considered "cloned" and the newly created jobs "clones". That's true for the context of this issue and the web UI uses the same terms in several error messages and the "Cloned as ..."/"Clone of ..." info.

or those jobs which are restarted via openqa-clone-job?

No. Jobs "restarted" via that script are actually not considered restarted/cloned within the context of this issue and also not by the web UI, e.g. the web UI would not show the "Cloned as ..."/"Clone of ..." info. Note that the openqa-clone-job does not support parallel and directly chained dependencies so it is not really helpful here anyways.

If we only rerun part of tests(a sub tree of the original test chain tree), will the 'cloning tree' graph be the sub tree only?

I'm not 100 % sure what you mean but this question likely falls under the points I've mentioned under "notes" within the ticket description. That means it isn't completely sorted out at this point.

Can we still view or operate the original tree after implementing this ticket?

I don't know what you mean with operate. Being able to view the original tree is the main point of this issue.

The graph is not just a graph, it is the actual chain, we can rerun jobs with the graph, do I get it correctly?

The graph is just a graph. It shows dependencies between jobs. It does *not* show the cloning/restarting chain. Showing the graph for jobs which have already been cloned does **not** mean you can restart these jobs now.

#5 - 2020-08-26 09:52 - Julie_CAO

Hi Marius, I think I get the meaning. Taking a ABCDE test chain as an example, currently, given we restart a sub chain, such as ABC, then only ABC are shown in the new created graph, DE are lost, DE are only showed up in the original graph. After implementing this ticket, in the same case, DE will be shown in the new graph as well even if they will not been restarted. The new graph is identical with the original one, ABCDE. But on the new graph, only ABC which have been restarted are allowed to be restarted again, DE are not allowed. Do I follow you exactly?

If yes, it helps in some extend. The test reviewers always see the correct test dependencies regardless any restarting.

Of course, what we are expected most is DE can be restarted in the new graph(or in the original graph, or in any other ways) as well.

#6 - 2020-09-01 16:02 - okurz

- Related to action #59969: *Display job dependency tab not only for latest jobs added*

#7 - 2020-09-04 09:28 - okurz

- Related to action #69979: *Advanced job restarting via the web UI added*

#8 - 2020-09-04 09:53 - okurz

Please understand that this ticket is prioritized with low prio and we keep it in the "future" target version, i.e. the SUSE QA tools team does not plan to implement this anytime soon (not within the next months or years). Given that the ticket has already received good and comprehensive explanations and implmentation suggestions it can be feasible for any outside contributor to implement. We are always happy to receive pull requests from anyone :)