

Item 358

git_comments:

git_commits:

1. **summary:** Merge pull request #852 from vorburger/FINERACT-922_SchedulerJobsTest-reset-world-as-found
message: Merge pull request #852 from vorburger/FINERACT-922_SchedulerJobsTest-reset-world-as-found make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)

github_issues:

github_issues_comments:

github_pulls:

1. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
2. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
label: test
3. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
4. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
5. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
6. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
7. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922
8. **title:** make SchedulerJobsTest preserve scheduler and job states (FINERACT-922)
body: FINERACT-922

github_pulls_comments:

1. Anyone willing to review this one? @nemux @awasum @xurror @ptuomola @trackcc LGTY?
2. **body:** The current failure here is totally unrelated to this PR, and another "flaky" test, see FINERACT-828.
label: test
3. It sounds like nobody seems to have any strong objections to this one, so I'd like to "self merge" it tomorrow.
4. The code itself looks good to me. I haven't hit this particular problem myself, so can't comment on whether it fixes the issue - but it does look like a sensible thing to do.
5. This looks clean..But i dont understand completely whats happening here. @vorburger Go ahead and self merge. You know what you are doing. Minor update..seems there is a merge conflict?
6. I have little knowledge about this but I trust the changes are OK.

github_pulls_reviews:

jira_issues:

1. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
label: code-design
2. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests

- description:** [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
3. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 4. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 5. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 6. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 7. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 8. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 9. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 10. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
 11. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
label: test
 12. **summary:** Improve SchedulerJobHelper to be more reliable and prevent flaky tests
description: [~ptuomola] pointed out in FINERACT-855 that the reason for at least some of our often Randomly Failing Integration Tests in FINERACT-850 is likely that the SchedulerJobHelper has serious issues. Improving that helper presumably will help for a number of flaky tests.
label: test

jira_issues_comments:

1. **body:** I spent some time looking into what [~ptuomola] pointed out in FINERACT-855, quote: {quote}This is a timing issue re scheduling the job: Quartz does not have time to process the job submission before we request the status of the job, and hence when we request the job status, it's neither completed nor currentlyRunning as expected by the code. {quote} Here are some notes for my future self, or anyone else interested: So e.g. {{SchedulerJobsTest}}, and 6 other scheduler based tests through their use of {{SchedulerJobHelper}} who's {{executeJob}} function seems to have been copy/pasted from {{SchedulerJobsTest}} (which we could clean up, and make {{SchedulerJobsTest}} also use {{SchedulerJobHelper.executeJob()}}), uses {{/fineract-provider/api/v1/jobs/\$jobId?command=executeJob}} to manually trigger running a specific named job. They then wait for a job's {{currentlyRunning}} to be false, and then check the (last entry of) {{jobs/\$jobId/runhistory}} to have a success status. I believe the point [~ptuomola] was making is that it's possible that from when Fineract's

{{SchedulerJobApiResource.executeJob()}} runs (Fineract's) {{JobRegisterServiceImpl.executeJob()}}, which calls {{org.quartz.Scheduler.triggerJob()}}, until when {{currentlyRunning}} switches from false to true and back to false, there could be delays and misses. The {{currentlyRunning}} seems to be set to true (only) in {{SchedulerWritePlatformServiceJpaRepositoryImpl.processJobDetailForExecution()}}, which is called by {{SchedulerTriggerListener.vetoJobExecution()}}. That **should**, in theory, be quick, but yes, we never know, so making that more robust could be a start I guess. But there also could be other reasons for failure, such as those OptimisticLockExceptions, causing jobs to fail.

label: code-design

2. Hi - yes, that's exactly what I observed with this particular test on my machine: The test called `executeJob()` to trigger the job and then retrieved the job status. But this happened so quickly that Quartz had not yet started the job. Hence `currentlyRunning` was false when the `SchedulerJobHelper` checked for job status. When `currentlyRunning` is false, `SchedulerJobHelper` assumes the job has completed and retrieves the job history. But in this case the job history was empty as the job had not even been started - which then resulted in a failed assertion (and failed the test). So in this case, waiting a few seconds before retrieving the job status fixed the test. Alternatively, as I mentioned in the earlier comment, the logic in `SchedulerJobHelper` could be enhanced to recognise this scenario and deal with it accordingly. And yes, I'm not sure how many other test failures are due to this (if any) - I only observed this issue for this particular test.
3. I gave this some thought, and propose [<https://github.com/apache/fineract/pull/817>] (requires FINERACT-926).
4. OK, so I am actually making some progress on this, and am now locally seeing FINERACT-957... It's currentl only in the logs (once you have FINERACT-956); I've re-opened FINERACT-858 to "propagate" it more clearly.
5. FINERACT-958 is the next error that pops up when FINERACT-957 is fixed... (I won't be able to fix FINERACT-958; help!)
6. FINERACT-992 is the error I'm currently seeing (and not FINERACT-958, although that may just be hidden for now, and come back after FINERACT-992 is fixed).
7. FINERACT-992 was a false alarm. The next step on this bug, after <https://github.com/apache/fineract/pull/817> and <https://github.com/apache/fineract/pull/813> and <https://github.com/apache/fineract/pull/812> have been merged, will be to start replacing, one by one, usages of the old `org.apache.fineract.integrationtests.common.SchedulerJobHelper.executeJob(String)` by my new `SchedulerJobHelper.executeAndAwaitJob(String)`, and see which ITs still pass (and investigate why they fail if they do), and which `@Ignore-d` IT (which use this helper..) will pass with this new & better helper - or why they still fail. After those 3 open PRs are merged (and also thanks to FINERACT-858), there will now be much clearer error messages than we used to have before. Let's put any findings into the open JIRA for the respective tests and not into this bug.
8. <https://github.com/apache/fineract/pull/956> <https://github.com/apache/fineract/pull/957> <https://github.com/apache/fineract/pull/958> <https://github.com/apache/fineract/pull/959>
9. <https://github.com/apache/fineract/pull/999> for FINERACT-857 pending. FINERACT-899 required next ([~ptuomola] expressed interest). <https://github.com/apache/fineract/pull/1000> to remove the "old" (bad) `SchedulerJobHelper.executeJob()` [FINERACT-922] #1000 Then this can probably be considered Done.
10. Resolving this as I was finally able to merge <https://github.com/apache/fineract/pull/1000> today. Many thanks again to [~ptuomola] for the (great!) collaboration on this topic.