2020-05-26 - VIVO Development IG

Date

26 May 2020

Call-in Information

Time: 11:00 am, Eastern Time (New York, GMT-04:00)

To join the online meeting:

- Go to: https://lyrasis.zoom.us/my/vivo1
- One tap mobile:
  - US: +16699006833,,9358074182# or +19292056099,,9358074182#
- Or Telephone:
  - US: +1 669 900 6833 or +1 929 205 6099 or 877 853 5257
  - Meeting ID: 935 807 4182
- International numbers available: https://zoom.us/u/aeANHanzED

Slack

- https://vivo-project.slack.com
  - Self-register at: http://bit.ly/vivo-slack

Attendees

⭐ Indicating note-taker

1. Huda Khan ⭐
2. Benjamin Gross
3. Brian Lowe
4. Michel Héon
5. Nicolas Dickner
6. Alexander (Sacha) Jerabek
7. Andrew Woods

Agenda

1. Updates - i18n sprint
   a.
   b. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
2. Collecting
3. Solr configuration needs updates: Unable to locate Jira server for this macro. It may be due to Application Link configuration.
4. VIVO Scholars - next steps
5. Tickets to
   a. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
   b.

key	summary	type	created	updated	due	assignee	reporter	priority	status	resolution

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

6. Who is working on what? - Are there active tickets in-progress?
   a.

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Future topics

1. Vitro JMS messaging approaches - redux
    a. Which architectural pattern should we take?
    b. What should the body of the messages be?
2. Incremen
    a.
    b. Unable to locate Jira server for this macro. It may be due to Application Link configuration.
    c. ion.
    d. Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Tickets

1. Status of In-Review tickets

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

Notes

Draft notes in Google-Doc

1. Updates - i18n sprint
    a. Benjamin working on changing the flags be the language and parenthetically the country
    b. Code looks good and does what was agreed upon by sprinters
    c. Andrew: Rebuild Vitro and not seeing it yet. Turned off caching as well as removed triples but not sure why not visible
    d. Benjamin: Is there another version of language selector ftl that exists in VIVO languages? Possibly missed something in a separate branch
    e. **Update :** issue was that pull-request branch is using earlier version numbers in pom.xml files
    a. Will do this with merge instead of rebase
    b. Less impact, b/c rebase would mean hashes would no longer match up and would need to do a re-clone
    c. Is there a timing that is better for this since don't want to be disruptive?
    d. Michel: No test case to evaluate the consequences of merge. Should have a solid test case before we try to merge
    e. Brian: Optimistic that the merge probably won't be problematic
        i. Andrew is also optimistic
    a. Next sprint: June 15 - 19
    b. VIVO-1839 - Text option for language selection drop down In Review
    c. Trying to keep sprint branch and master as close as possible
    d. Matthias' co-worker Dominic will be joining in next sprint
2. Collecting use cases: first-time, everytime, filegraph
    a. If language are in first-time and others are in everytime. In development phase, if we do not want to drop the tdb directories, have to move core ontologies into everytime and have to move all the labels ontology into everytime. After development, when we want to push to production, have to move everything back to first time.
    b. Simple mechanism to erase tdb to remake the tdb each time you make changes in the label ontology
    c. Because default installation of VIVO suggests SDB, that will make it complicated for developers working on languages
    d. Benjamin: Brian's summary in slack channel was good (Huda: I think this is the relevant summary)
        i. "I haven't caught up on this whole thread yet, but the three categories are: "(1) firsttime": files whose contents should be loaded only when the application is first installed (or triple store is reinitialized), should only be updated automatically when the upgrade process runs during startup of a new version of VIVO, and whose contents are expected otherwise to be edited through the GUI. (2) "filegraph": files whose contents should always be checked/loaded at each startup of VIVO so that there is a graph

maintained in the content triple store that matches the contents of the file. (3) "everytime": files of application-related RDF data that are loaded into in-memory models every time that VIVO starts, but which are **not** also persisted in the triple store (because it is not expected that anything other than the VIVO application will be querying them). Each type of RDF content (abox/tbox/applicationMetadata/display/auth) supports at most two of these three modes, and some (like application metadata, if I recall correctly) support only firsttime. Adding things for additional languages to "everytime" (where available) instead of "firsttime" prevents their editing by an end user and seems counterproductive to the goal of internationalizing the editing interfaces. A better longer-term solution is probably either to enhance the firsttime processing so that the upgrade-style merge logic can integrate new changes from the filesystem without emptying the triple store, or to retain both the filesystem-based defaults plus end-user edit deltas and always query the two such that the latter data overrides the former."

e. Brian: As Benjamin said, since the things you loaded in first time and things you've edited are all combined in the graph, if removing files, then are reverting back to first-time (only) setting
f. Michel: Setting this up as a development property
g. Brian: Priority more about installation for non-developers as opposed to development-centric features
h. Benjamin: putting in a flag in runtime.properties.
i. Andrew - use cases
   i. "As a VIVO administrator, I want to install a new language after having run my VIVO for some time" (i.e. existing VIVO)
   ii. "As a VIVO developer, I want to implement new languages"
   iii. "As a VIVO non-developer, I want to implement new languages"
   iv. Mechanism to flush the appropriate triples to resolve use cases 2 and 3
   v. Use case 1 would not be resolved by the "flush all the triples" approach
j. Brian: The two graphs editable through the GUI (kb2 and asserted tbox). Problem is default in first time go into two graphs. Any changes in GUI are applied to those two graphs.
   i. If nothing in asserted tbox b, brings in from asserted tbox a (hope I'm getting this right)
   i. Brian: Multiple versions and then assessing which to use at query time. Long run, should work well. But quite a big change in the current system.
   i. Before we were using named graphs/quad stores so some of that GUI stuff is based on older mechanisms
   ii. Split to have Kb and Kb2 - first time goes to Kb2 and any triples added through the GUI are in KB, so can keep track of provenance of what is added through the original file load
   iii. Reload behavior: could go through and if something has a label in a given language and doesn't pull in triple from first time (assuming have set)
   iv. Could do this everytime on startup but would take longer, so instead could be looking at flag/property in runtime.properties
   v. Andrew: Splitting into two graphs so don't have to remove all UI-based changes. Additional logic for querying both graphs
k. Andrew: Brian, please put in your own notes or document suggested approach for later discussion:)
a. Conversations around first time and everytime graph files
b. If we can identify use cases for how we would like VIVO to work as it relates to these different triples
c. Andrew: requires removal of tdb models. If upgrade VIVO ontology, first time recognizes the upgrade and pulls in the first time files again. Issue is, in terms of configuring languages, have to decide languages when you first start up VIVO.
d. Need to have clarity across the team regarding what these first time/everytime files do. Link goes to first-time, everytime, filegraph wiki page. Need to understand what these files do so can see what we'd like the behavior to be.
e. Michel: config is ok but need to know how to use it during development.

3. Solr configuration needs updates: VIVO-1866 - solrconfig.xml included with vivo-solr is missing configuration from pre-1.11 versions Open
   a. Benjamin: Until noticed caching wasn't working, didn't really notice issues.
   b. Brian: have been seeing alltext field which is the catch all (where all the things are concatenated together)
      i. Queries across context nodes. All of that text was concated into alltext field so that search had a hit on publication title, also got the people who were authors for that publication
      ii. Now no longer seems to be the case. Hits only for publication title but not people anymore
      iii. Number of search hits is dramatically smaller
      iv. Alltext field has a bunch of empty "nothings" - this is what is happening in 1.11 but not in 1.10
   c. Huda: Need to keep digging but do see various tests up through 1.9 that focus on search indexing behavior
   d. Andrew: alltext issue needs further review
   e. Need to compare 1.10 with 1.11 to see what differences there are
   a. Documented solr configuration is a bit of out of date
   b. Huda may have updates on Cornell specific things going on
   c. Benjamin created a diff between the 4.X Solr config and the newer Solr config (7.x). Note of caution: diff is reversed.   VIVO-1866
   d. Andrew: Two approaches: bringing in as much of previous solr configs as possible, or actually documenting and understanding what to keep and why. Otherwise, not know which changes need to stay.

Actions

-