2020-04-07 - VIVO Development IG

Date
07 Apr 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

Attendees
- Indicating note-taker
  1. Don Elsborg
  2. Andrew Woods
  3. Huda Khan
  4. Benjamin Gross
  5. Ralph O’Flinn
  6. Brian Lowe
  7. Michel Héon
  8. Alexander (Sacha) Jerabek
  9. William Welling
  10. Mike Conlon
  11. Steven McCauley

Agenda
1. Sprint updates - i18n
   a. Discuss potential methods for external consumers like VIVO Scholars to query/harvest data from an i18n VIVO
   b. Sprint details
      i. Test plan
      ii. New GitHub repo for regression/Selenium tests?
   iii. 2020-04 - Sprint Stand-up Reports
2. vivo-tech emails
   a. Missing N3-PP writer and RiotException - VIVO-1761 - Failure when using toString() on AdditionsAndRetractions.java - CLOSED
   b. How to remove people from Vivo - two solutions are being suggested... could someone create JIRA tickets?
      i. > a nice feature in the UI would be a deletion option that allows you to delete the primary object, plus whatever else is created via the UI when you create a new object. It could re-use the logic in the form controller
      ii. > Another possibility would be an automatic deletion of orphan objects: vcards without individuals, publications without authors, and for us membership without members. This could take place synchronously or asynchronously, through a script
3. Vitro JMS messaging approaches
4. Moving tickets forward
   a. VIVO-1658 - Preparing for Production regarding About Page and the Support Link - REOPENED - back on your plate, Mike Conlon
   b. VIVO-1761 - Failure when using toString() on AdditionsAndRetractions.java - CLOSED - need 2 reviewers
5. Who is working on what? - Are there active tickets in-progress?

<table>
<thead>
<tr>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| VIVO-1888 | i18n: Supporting different countries with same language | Michel Héon, Andrew Woods | IN PROGRESS | Unresolved | May 26, 2020 | Jun 01, 2020 |
| VIVO-1720 | VIVO Scholars test dataset | Julia Trimmer, Andrew Woods | IN PROGRESS | Unresolved | Oct 14, 2019 | Oct 28, 2019 |
| VIVO-1698 | Add FAST vocabulary lookup to VIVO | Don Elsborg, Don Elsborg | IN PROGRESS | Unresolved | Aug 06, 2019 | Aug 08, 2019 |
| VIVO-1684 | Create multi-stage Dockerfile for VIVO 3 tier custom build | Don Elsborg, Don Elsborg | IN PROGRESS | Unresolved | Mar 21, 2019 | Mar 21, 2019 |
| VIVO-1660 | Create fr_CA language files | Kitio Fofack, Kitio Fofack | IN PROGRESS | Unresolved | Dec 06, 2018 | Dec 06, 2018 |
| VIVO-1651 | About page should have link to registry and other VIVO support info | Mike Conlon, Mike Conlon | IN PROGRESS | Unresolved | Dec 04, 2018 | Nov 01, 2019 |
| VIVO-1645 | [Experiment] Product Evolution: Elasticsearch/GraphQL/React Static | Greg Burton, Greg Burton | IN PROGRESS | Unresolved | Dec 03, 2018 | Dec 13, 2018 |
| VIVO-1597 | Document Triple Store/Search Index workflow | Steven McCauley, Steven McCauley | IN PROGRESS | Unresolved | Sep 21, 2018 | Nov 01, 2019 |
| VIVO-1450 | Document the Maven 3-tier build process for upgrades | Don Elsborg, Mike Conlon | IN PROGRESS | Unresolved | Feb 21, 2018 | Dec 03, 2018 |
| VIVO-1446 | Create Vitro Application Configuration Owl file | Brian Lowe, Mike Conlon | IN PROGRESS | Unresolved | Feb 19, 2018 | Mar 13, 2018 |
| VIVO-1443 | Decouple VIVO build from installation | Andrew Woods, Andrew Woods | IN PROGRESS | Unresolved | Feb 07, 2018 | Apr 22, 2020 |
| VIVO-1439 | Add command line help option to jena2tools and jena3tools | John Fereira, Mike Conlon | IN PROGRESS | Unresolved | Jan 31, 2018 | Jul 11, 2018 |
| VIVO-936 | A glossary is needed with all VIVO /VITRO templates, their locations, and purpose | Mike Conlon, Don Elsborg | IN PROGRESS | Unresolved | Jan 05, 2015 | Apr 21, 2018 |
6. Incremental development initiatives

a. **VIVO-1688** - Separate VIVO ontology into own repository [OPEN]
b. **VIVO-1751** - Externalize Asset Store [OPEN]
c. Integration test opportunities with the switch to TDB - requires startup/shutdown of external Solr .. via Maven
   i. **VIVO-1752** - Instantiate Solr core via API [OPEN]

**Tickets**

1. Status of In-Review tickets

<table>
<thead>
<tr>
<th>T</th>
<th>Key</th>
<th>Summary</th>
<th>Assignee</th>
<th>Reporter</th>
<th>P</th>
<th>Status</th>
<th>Resolution</th>
<th>Created</th>
<th>Updated</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>T+</td>
<td>VIVO-1885</td>
<td>i18n-regression-test documentation</td>
<td>Michel Héon</td>
<td>Michel Héon</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>May 25, 2020</td>
<td>Jun 19, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1873</td>
<td>Internationalize SPARQL query page</td>
<td>Andrew Woods</td>
<td>Benjamin Gross</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>May 20, 2020</td>
<td>Jun 09, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1863</td>
<td>Menus are presented in French Canadian in the en_CA context.</td>
<td>Michel Héon</td>
<td>Michel Héon</td>
<td></td>
<td>IN REVIEW</td>
<td>Fixed</td>
<td>May 12, 2020</td>
<td>Jun 19, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1839</td>
<td>Text option for language selection dropdown</td>
<td>Benjamin Gross</td>
<td>Benjamin Gross</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Apr 17, 2020</td>
<td>Jun 19, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1801</td>
<td>Selenium regression test</td>
<td>Andrew Woods</td>
<td>Michel Héon</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Apr 15, 2020</td>
<td>Jun 01, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1734</td>
<td>Remove one-time permission in ORCID language strings</td>
<td>Christian Hauschke</td>
<td>Christian Hauschke</td>
<td></td>
<td>IN REVIEW</td>
<td>Fixed</td>
<td>Nov 08, 2019</td>
<td>Dec 09, 2019</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1731</td>
<td>Updates to maven plugins and code dependencies</td>
<td>Ralph O’Flinn</td>
<td>Ralph O’Flinn</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Oct 25, 2019</td>
<td>Mar 04, 2020</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1730</td>
<td>Dockerize External TPF Server</td>
<td>Andrew Woods</td>
<td>Andrew Woods</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Oct 23, 2019</td>
<td>Oct 30, 2019</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1728</td>
<td>Document how to install / configure externalized TPF</td>
<td>Andrew Woods</td>
<td>Andrew Woods</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Oct 22, 2019</td>
<td>Nov 12, 2019</td>
<td></td>
</tr>
<tr>
<td>T+</td>
<td>VIVO-1723</td>
<td>Externalize TPF Server</td>
<td>Andrew Woods</td>
<td>Andrew Woods</td>
<td></td>
<td>IN REVIEW</td>
<td>Ready for Test</td>
<td>Oct 14, 2019</td>
<td>Mar 11, 2020</td>
<td></td>
</tr>
</tbody>
</table>
### Notes

**Draft notes in Google-Doc**

**Sprint updates - i8n**

Sprint started Monday, this week and next week. Kickoff meeting happened.

Re: consuming i8n VIVO data for scholars, scholars team needed to determine how that would work with Scholars. How would tags work?

Andrew - presumably this is related to triples in store and how this data would be accessed?

William - could be done a few ways.

- Address sparql queries for each language. This would be redundant
- Another way is to have multiple harvest stages, one for each language, this would then put the output into separate solr indexes, one for each language.
- Biggest concern is how to represent this in a discoverable way. In Solr and how the UI would access the language data. William doesn't see a problem in the triple store itself.
- Ralph - so how to allow a choice if there are languages. Should Scholars discover show the languages the same way as VIVO core

Andrew - so the issue might be more with Scholars displaying language data.

Michel - linguistic data needs to be put in the SPARQL query. So when the users are doing certain actions, the language context is placed in the query. Not sure how SOLR is working with this at this point? He says SOLR is working fine and it knows what the language is. Easier to talk about architecture.

Don - so if you are doing a search and you're in the French context, does the search only return French?

Michel - not quite sure how SOLR is used in the architecture.

Brian - doesn't believe that VIVO Solr is working with languages.

Andrew- so need to know how SOLR search is operating.

Brian - all lang literals placed in ALLTEXT, so if searching esp using a language term, it does return because it's in the index. BUT, all languages are jumbled together. The shortview will retrieve the label of the individual from the sparql query. So will get the language context from the search.

Michel - the Germans did the language choice functionality.

Andrew - Brian's explanation of language search results being context aware seems like the right answer. So quasi results but not architected for language awareness.

Brian - there used to be a problem in older VIVO where things looked like it worked, but one part of the interface where there were inconsistencies because labels might not have been queried right.
Michel - each piece of text needs a linguistic label. It's a clear request and we have the right text because the query is based on the linguistic context. For users who need multi-lang, each piece of text needs to have a linguistic tag. True for vcards and true for every other ontological data element.

Andrew - The lang tags assignments are done in the UI with this sprint

Michel - yes, each text has a tag which is added to

Lots of discussion on ALLTEXT contents vs the labels returned by search. Fill this in

So search ALLTEXT issues need to be identified and a pattern developed to accommodate lang tags in search.

William - for the case where a lang context is picked .. static text should be changed according to context. So if the dropdown lang picker is set,

1. UI - when lang is picked - all tags get changed
2. API - does that respect the lang header/tags
3. If there's a lang header, does the triple store respect this and when/how should SOLR

Andrew - certainly static text and also the triple store

William - if lang header is set, does VIVO only return triples for that lang tag in header.

Brian - this used to be the case, but not entirely sure anymore. Needs to be checked.

William - if dropdown is selected, for French, should only French be returned. A standard should be set and applied. If this paradigm is set in Vitro, it should also be used for Scholars. So everything is looking at the appropriate place for the lang tag - eg browser header.

Michel - the mechanism on how to find the linguistic context isn't important, if we send a sparql request we need to know which language we're in.

Brian - general principle for end user, we can't get rid of the language drop down and can't count on users changing headers.

William - if researcher is editing VIVO, if they want to write abstract in EN, they have to select linguistic context, then go into the editor and write abstract. Then if he wants to switch to FR, they have to save, then go back and change language context for FR and go back to the abstract editor and write it in FR and save. Good argument to have a drop down language override on individuals

Action - investigate recommended patterns for SOLR to support language contexts.

Lot more questions, optimization, language context overrides, honoring headers, etc

Agenda 2a

- 1761 - look at PR please

Agenda 3 - JMS messaging, link and slide

https://docs.google.com/presentation/d/1dVRoE8xy4le1-TDikM5AwjKT5BmRavhCM4Qn-Nd3HE/edit?usp=sharing

Andrew wants us to gain consensus on which messaging approach to pick.

How to capture the information for code discovery, javadocs are a natural choice, but this also requires publishing the docs. When people investigate code, is there another way to surface the discovery?

Brian - a diversion might be to document all of the methods. In the rdf display model, there's a whole set of properties on document findings. Can create appropriate RDF entities. This is in the RDF configuration ontology. This might be more useful then docs for each method.

Mike - so who is setting and using this.

Brian - he used this to document configuration properties on what the search indexer is doing is a good place to start

Huda - The n3 for search indexer and config, and there's the code for the search indexer

Mike - found dozens of rdf configuration files. How to start this document.

Brian - start with just the search indexer. Jim started this, if you find an example in the rdf file you can grep into java to find the methods.

Action: Brian will look for or draft documentation of the important features of URI finder and document builder configuration in the display model. Understanding these features is very useful for understanding the important components of VIVO's search indexing.

William - difficult to find the things that pinpoint what has changed. A design needs to happen with the pinch point on when a triple is changed, that needs to be identified.

Brian - rdfsServiceFactory is there and there is a single place in here where you can register a listener. This might not be hard to see where ...

Queues of URI's that have been changed, then actions happen on the queues. So whole complex cascade of what happens. So RDFS and RDFSServiceFactory. There should be a place where listeners register to hear about changes.
How to register to listen, a java class that runs during startup. It would be logical to set this up in the applicationsettings.n3, this would be better then placing it in the java class.

Andrew - messaging would be registering a service for the Factory queue.

Brian - current listening is only at the triple level. Are we happy with this only happening in the triple level or should we synthesize a document change and send those messages to clients.

William has a PR to Vitro that listens to data source in TDB and it detects data source changes with rdfDelta. But if externalized triple stores are the approach, then the listener “pinch point” should be above the actual data source.

Actions - how to find the pinch points and what pinch points are we listening too.

Brian - high level goal on what we want to achieve with this. Would be nice to offer a messaging solution that allows people to not think about RDF. A standard mechanism that does the complicated things related to what changed and send a changed document out.

Don - so can we plug in Williams Tamu discovery harvester to a “pinch point” listener.

William - will need to still identify what has changed and how the shape of the changes correlate to the documents that would be created with something like the Scholars Discovery harvester.

**Actions**

- [ ]