

2017-05-16 Meeting notes

Date

16 May 2017

Facilitator: Andrew Myers

Notetaker: LaRita Robinson

Attendees

- Jennifer Lindner
- Bess Sadler
- Lynette Rayle
- LaRita Robinson
- Aaron Collier
- Andrew Myers

Agenda / Notes:

1. Report on progress, check on board: <https://github.com/projecthydra/projecthydra.github.io/projects/1>
 1. Coding Style - Lynette
 1. linked to code comparison to branch covering coding styles
 2. going to link to sample configuration style
 1. <https://github.com/projecthydra-labs/bixby>
 2. Effort may have stalled at the end of the dev congress on this issue.
 3. Good to have people look at it. (Can't load it up and look at changes to see it formatted, but can read code changes in github.)
 4. Scripts to copy the configuration
 1. might want to provide a configuration gem to "hydrify a project"?
 2. might be good to just store in hydra-head
 5. Missing sections:
 1. best practice on general "these are good ways to program ruby in context of the hydra community". Unpack what "coding well" means.
 2. Patterns that you see in the code
 6. will update issue with a checklist to determine "done-ness".
 2. Stack overflow channel - Lynette
 1. Not as easy as originally thought. Will continue to look into this.
 2. Is there a way to have hydra automatically go to our slack channel?
 1. not a good idea now because hydra means more than just our community.
 2. Good to wait for branding so we don't have to do tag cleanup.
 3. People might be more open to this if it can be integrated into slack.
 4. Added "definite done" point question
 3. Issue template - Aaron
 1. done
 4. Doc.new - Jennifer
 1. When typed in slack, opens the issue template automatically via slackbot, to create a new issue regarding documentation.... this part is done
 2. gives back the link response. Maybe could be made more configurable?
2. Other discussion

Need another ticket "how to ask for help", etc?

 1. Is there something on duraspace already? What else is available? Should it be moved or copied to our documentation site? Will make as a new issue to move it

What other documentation should be moved as well?

 1. if it goes beyond a specific repository, it shouldn't be on that repository. Developer's guide in hyrax goes beyond hyrax only and should be moved. Should also be an issue.
3. We want to be a single stop for developer documentation, or at least an obvious first place.

Not all documentation needs to live on our site, but should have links to it as ways to know it exists. How much should be migrated vs linked?

 1. Having it all in one place is more obvious than having it dispersed.
 2. We don't want to say "don't write any docs on the wiki". We want to provide some sort link to any documents which are there.
 3. Don't want to spend energy and scope creep to migrate all documentation but we do want to insure that we link to what exists.
 4. We need to do a better job to capture what all of the sources of other documentation are.
 5. Initial goal is to capture... we want to set up a framework to move toward.

6. Bad documentation is bad because it's out of date or people don't know where it is. We can't solve the "out of date" problem, but we can minimize the maintenance cost of keeping it up to date. This happens by making it easier to find.
7. If we cultivate a good stack overflow area, it may help with the "keep it up to date" problem. If someone tries something and doesn't work, we can tag as deprecated. Helps us capture what is out of date and has fallen off radar.
8. Demarcation line is "if it's a gem that is encapsulated, keep it with gem". If general practices, bring it into our centralized repository. Allows us to download and search.
9. Goal to set up a framework, not to do migrations now. So our process should be to create new tickets for anything stale, broken, bunch of lies. Tag appropriate person to update bad documentation. "Done" looks like: delete the old stuff, make new in the new documentation site.

Facilitator's role:

1. Should we have the same facilitator for each meeting or pick one for each week?
2. might be nice to rotate the facilitator
 - Agenda making is biggest piece of the work
 1. at end of every meeting, we'll design an agenda-maker who will then run the next meeting.
 2. have standing items on agenda so we can just do a cut/paste to cover the main items
 3. Everyone can add items throughout the two week period
4. Can rotate meeting-runner, but good to designate other roles: communications person, github ticket backlog groomer, agenda creator, etc. Make sure the offline work gets done without it all relying on one person.

Notetaker's role:

1. Create the agenda for the next meeting immediately upon finishing the notes from current meeting.

WG goals:

1. leave documentation where most appropriate
2. provide links from first location (documentation website)
3. establish protocol for fixing documentation through this working group.

What should we list as action items, and how do we move items along?

1. Items being worked on aren't necessarily going to be done by the next meeting. We should review the status of in-progress items, and also look at ready items from the project board.
2. Good to dedicate some hours to work on issues, maybe pair up with people on slack to work on issues

9. Standing items for agenda:

1. decide agenda-maker/next meeting facilitator, and note-taker
2. review last meeting's action items (in progress tickets)
3. What should we move to in-progress to discuss next week?
4. groom the backlog

3. Pick facilitator / notetaker for next time: (oops! we missed this! Will be decided at beginning of next meeting)

1. Facilitator:
2. note taker:

Action items

- Drew will send out a doodle poll regarding pairing up on tickets in ready or in progress column to push them to done. Jennifer has the most time to allocate currently, so we can try to pick a time that works to pair up with her.
- LaRita will create agenda page for next meeting