Blacklight Hydra Committers Call 2011-11-21

Participants

- Jessie (Stanford)
- Matt Zumwalt (MediaShelf) - moderator
- Justin (MediaShelf)
- Joe (UVa)
- Naomi Dushay (Stanford)
- Bess Sadler (Stanford)
- Chris Beer (WGBH)
- Bill Parod (Northwestern) - notetaker
- Others...

Agenda

1. Select Moderator and Notetaker
   a. Matt Zumwalt moderator; Bill Parod notetaker
2. Call for Additional Agenda Items
3. Blacklight Project
   a. strategic directions
      i. 3.2 will be out in the next few weeks: Multiple configs per controller;
      ii. SASS Refactor, abstracting out the stylesheet for rendering;
      iii. Streamlining customizability, making core leaner / modules that provide more advanced features;
   b. release planning
      i. Master git branch should always be runnable;
      ii. Feature branches are used for development and fixes;
      iii. Try to do semantic versioning / debatable; Try to insure that patch releases don't break anything; Though because any method can be overridden, it's difficult to claim that a new version is backwards compatible.
      iv. Perhaps over time, Blacklight can start declaring private methods to protect backwards compatibility, facilitating semantic versioning.
   c. overview of configurability, intended overrides, etc
      i. It's a Gem;
      ii. Worked on the helpers and controllers to make them more "Railsy";
      iii. Catalog behaviors are all in a module;
      iv. Can pull in Blacklight functionality as needed; CatalogController renaming can be accomplished in router;
      v. Blacklight provides more straightforward hooks for setting search params and "before" filters based on current controller's request. Hydra uses that to inject access control into the searches for gated discovery;
      vi. Helpers (View helpers) are now organized as modules that you can include in your helpers;
      vii. Justin reworked Hydra's helpers to work that way; Motivated by wanting to make it easier to override Helper methods;
      More explicit about what is being loaded and when.
   d. how to do Blacklight patches
      i. Create Jira tickets
      ii. Make pull requests
      iii. Attend IRC channel to answer questions.
      iv. What's the best way to contribute plugins?
         - Same issues as writing and sharing a gem. When should it be in a gem and when should it be in blacklight?
         - Needed hooks to make a gem possible probably should go in Blacklight;
      v. If you're making a plugin and have to fight Blacklight in some way to make it happen, that's a candidate change to core.
4. Hydra Project
   a. strategic directions
      i. Up until Rails 3, the Hydra Rails 2 use of Blacklight:
      ii. Start with Blacklight - add to that edit views and assets controller to update, create, delete assets (Fedora objects);
      iii. Edit views are mitigated through Cataloger; Asset CRUD requests go through a single controller;
      iv. Rails 3:
      v. With ActiveModel support if Rails 3. Now ActiveFedora objects behave more like ActiveModel objects; Using ActiveModel APIs.
      vi. Moved Hydra towards controller/views per model (create, edit, show, update, delete) methods
      vii. Blacklight's role is now confined to searching rather than also providing the application's broader MVC framework.
      viii. Makes the code more readable and is more "Rails-ish" and recognizable to Rails developers.
      ix. Facilitates workflows - multi-step interactions - by allowing each content type to have its own controller
      x. Blacklight is now the "Solr controller"
   b. release planning
      i. Moving towards quarterly release; patch releases "weekly" if there are new patches that week;
      ii. Patch releases are just bug fixes; no new features;
   c. Blacklight usage that blacklight committers should be aware of
      i. For release planning, should we consider synchronizing releases?
      ii. Rails 3 timing made sense. If Hydra is putting a major release every quarter, perhaps coordination with Blacklight, we can prevent lags between Hydra/Blacklight major release.
      iii. Knowing in advance what changes are coming in the other platform makes it easier to react and keep your platform compatible.
5. Hydra Project's Pinch Points
   a. please put specific pinch points here

   - Good things that have helped Hydra - putting helper behaviors in lib has really helped Hydra
• If there are controllers that should be in its own module, let Blacklight know.
• The other direction - pulling a controller out of a module - is also possible, though more difficult, but let Blacklight know when that’s useful.

• Overriding Views; Each function as gems, provide partials rather than overrides.
• Easier to provide your own layouts without having to undo/override Blacklight partials; continually with each new head
• The SAS makes it easier to modify as well.
• In writing thorough documentation on writing a Hydra Head and customize, have covered some Blacklight documents. Will notify Blacklight folks for pickup / ownership...
• Blacklight documentation is editable. Perhaps Hydra should add/edit/reference it.
• Want to encourage the Hydra folks or other Blacklight adopters to let Blacklight know when something needed is missing; Or write it up and put in a pull request.

1. Blacklight project’s action items: What configurations, hooks, API, documentation should be added?
   a. None

2. Hydra project’s action items: What in Hydra code should change to better leverage Blacklight?
   a. Jessie and others: what in Hydra code should change
      i. configurations?
      ii. helpers
         (e.g. refactor code to use x approach instead of monkey patch for feature y)

• Distinguish framework from heads:
• Here’s Hydra the framework
• Here’s the Hydra the MODS editor gem
• Here’s the Hydra gated discovery gem/plugin
• Will discuss separation of MODS editor from core head in next committers call.
• Hydra Solr configuration needs work and attention. Modifications are needed to use it in other contexts/environments.
• Will schedule a joint Blacklight Hydra call like this as part of release planning