Workspace: Hyrax feature spec reorganization

Intention regarding feature specs

There is an intention to refactor feature specs. This plan includes:

1. Making feature tests not run as part of the CI build (to speed it up)
2. Expanding the number of feature tests (to increase coverage and robustness)
3. Creating a periodic (nightly? weekly? etc.) CI build that runs all the feature specs
4. Obtaining a commitment to monitor these builds, and troubleshooting and fixing them as they break.

This page exists as a workspace for collecting thoughts and questions, and for documenting the current status of these specs.

Designing quality feature specs

What is the goal of feature specs?

The purpose of feature specs is to provide integrated testing. To insure that the integrated behavior is as expected, feature tests allow for a walk-through of the actions a user would be taking.

Individual pieces of the code should each have their own specs elsewhere, but those unit specs use a lot of mocked behavior to limit the tests to just that portion of the code. The integrated tests insure that when we put everything together, it works and looks as expected.

The features that we consider to be central to the application should be covered by feature specs. As new features are being developed, we need to decide what the integral functions are, and add the appropriate feature specs.

What should be our standards for feature specs?

1. Should specs mirror QA testing process?
   a. While this wouldn’t eliminate the need for QA, as that does significant cross-browser testing, it would make the first few rounds of QA testing quicker and easier.
2. Should specs minimize number of unique sections to avoid rebuilding objects multiple times?
   a. each time we have to rebuild the objects, it lengthens the run time of the specs
3. Integration testing generally involves taking a number of sequential steps through a process. Are there valid feature specs which don’t include sequential steps?
   a. Could a test without sequential steps be just as effective as a unit test? If so, is there a reason to repeat it in feature specs?
4. Should feature specs mimic the real world, or do we still want to short-circuit some aspects?
   a. How much should feature specs include js: true?
   b. How much should feature specs include :with_nested_reindexing?
   c. How often do we do :clean_repo?
5. What are the important things for integration testing? Many items could be tested in unit tests... should they be included in feature tests as well?
   Examples:
   a. pagination
   b. disabled buttons
6. How do we document the intent of individual feature specs?
   a. We use YARD standards for code. Should we have an expectation that specs are documented beyond the rspec context, describe, and it statements?
   b. Currently, many of the specs do not clearly document their intended purpose. Without knowing exactly what we intend to test, changes to the code may cause a test to be inadequate even though it still passes. For example,
      i. the collection feature spec “show work pages of a collection shows a collection with a listing of Descriptive Metadata and catalog-style search results”
         1. builds solr documents for 12 collections,
         2. displays the collection show page,
         3. tests that the page CSS includes ".pagination".
      ii. However there are three possible pagination sections on the page, and there are no wrappers to differentiate between sections.
      iii. The spec doesn’t use :clean_repo, so there is no guarantee that we are actually finding the pagination of the works section.
7. Should pull requests reviews have any specific guidelines regarding feature specs?

How do we connect feature specs and release testing?

Can use cases be built during development which can then feed into both feature specs and release testing spreadsheets?

Related release testing pages

- How to coordinate testing for a release
- Hyrax Testing and Release Process

Analysis of existing feature specs

- Collection feature specs
- Sample feature specs
• Add more case pages here