Abstract
Last week I reported a small change to the Rogue Scholar science blog submission form. By asking for the homepage URL of the blog instead of the feed URL, I hope to make it easier for users to register their blog. At the same time, I fixed a bug in the submission form, caused by an issue with the database backend. Unfortunately, that bug fix didn’t work as expected.
Last week I reported a small change to the Rogue Scholar science blog submission form. By asking for the homepage URL of the blog instead of the feed URL, I hope to make it easier for users to register their blog. At the same time, I fixed a bug in the submission form, caused by an issue with the database backend.

Unfortunately, that bug fix didn't work as expected. There were again issues with submitting new blogs, or with updating the information in the form. As a quick fix, I have now disabled the automated background process that automatically syncs the database record to the search index. And the submission form now contains an email link at the bottom for users who run into problems or have questions:

Before investigating the underlying issue further, I decided to add end-to-end testing of the Rogue Scholar user interface using the Cypress open source platform. This makes it easier to address and test more complex issues with the Rogue Scholar user interface such as

- forms (currently blog registration and search),
- mobile views,
- translations (Rogue Scholar is available in six languages),
- download of metadata and/or content in multiple formats.

More importantly, these tests can be fully automated and starting today run on every new deployment of the Rogue Scholar platform to production or staging.
This allows better handling of regression errors like this one (functionality that used to work and then breaks). These kinds of errors are a particular risk when refactoring a service (e.g. using a different library for the same functionality), something that young services like Rogue Scholar frequently do. The initial set of tests doesn't yet handle the blog registration workflow, but I plan to add that coverage in the coming weeks – and properly fix the underlying issue. The Rogue Scholar API already has good test coverage using the pytest framework, and the combination should become a solid foundation for Rogue Scholar going forward.

References