2016-12-01 DSpace 7 UI Working Group Meeting notes

Date

01 Dec 2016

First Hour: Angular meeting

Attendees

- Art Lowel (Atmire)
- Andrea Bollini (4Science)
- Tim Donohue
- Mark H. Wood
- Martin Walk
- Giuseppe Digilio (4Science)
- William Welling
- James Silas Creel
- Ying Jin
- Matteo Perelli
- Monika Mevenkamp

Starter Project

- Art tried out the standard universal starter project, angular-cli's universal branch and universal-cli.
    - angular-cli's universal branch get's updated not nearly often enough, current version is several months old so that's out
    - universal-cli is maintained by a single developer, there's not enough of a guarantee it will remain active. It's also still in alpha state so the risk of breaking changes is high.
    - The standard universal starter project does have a healthy following. It's also more configurable than the cli projects
- The project was started on https://github.com/DSpace/dspace-angular based on universal starter
    - The demo components were removed, a few things were tweaked to better match the style guide and tslint and codelyzer were added
- William mentioned that:
    - The universal starter project has angular-cli as a dependecy. It should be possible to use the angular-cli generators in this project.
    - The universal starter project also contains a beta version of an official meta-tag service. i.e. something that lets you modify <meta> tags in the <head> which is needed for google scholar support. We had to write our own in the prototype, but it should be better to start from the official version this time around.
    - When building for production we'll need to use ahead-of-time compilation as it provides massive performance gains and prevents a number of bugs

Redux & JSData

- Redux looks promising, and the @ngrx redux module for angular 2 integrates nicely with rxjs.
- It makes testing easier, but can make trivial changes more cumbersome: if you want to add a new state change you'll need to create or modify three separate files.
- When choosing whether or not to adopt a new framework or library we should always ensure that it works well with universal, to ensure good SEO support
    - So we'll need further testing to see how well it meshes with universal
    - We'll also want to test the combination of JS Data and universal

Browse work package - high level tasks

- Implement the classes (models, adapters, serializers, a store, …) required to interface with the REST API for collections, items, bitstreams and metadata (not communities as we decided last time to leave the concept of communities out of the UI)
    - It should have caching,
    - but also take in to account the fact that objects will be returned from the new REST API partially. e.g.
        - to show an object in the trail, we'll only fetch the title and an ID
        - To show them in a browse list, we'll need a little extra info
        - To show an item page for example, we'll need everything
- Port the components for collection home pages and item pages.
    - Adapt them to suit the new structure, backend, CSS framework when needed.
- Port the metatagservice to ensure the correct metadata is in the head for google scholar.
    - We should investigate the metatagservice in the universal starter project first to see if it can do all we need.
- Write missing tests for ported code.

- The initial goal will be to create a like for like version of the current UIs: by which we mean:
    - everything that's possible with the current UIs should also be possible with ours

    - we don't want to put a lot of effort in implementing new features
- That doesn't mean that everything should look and behave exactly as it does now
- But take your inspiration from the current UIs if you don't have a particular reason to do it differently

How to contribute

- Prerequisites
    - You'll need to be in the dspace-angular github team to get the required access. Send your github username to Art Lowel (Atmire) to get an invite.
    - Fork the repository on Github
- Workflow
    - Have a look at the waffle board
    - Take an issue that's in the ready section and has nobody assigned to it
    - assign yourself
    - When you start working on it, move the issue to the "in progress" section
    - Work on a separate branch for the issue on your fork
    - When you're ready, fire a pull request
    - in the comments of the pull request, write something akin to "this PR connect's to #{the ID of the issue}". That way the issue will be moved automatically to the review column.
    - When at least two people have reviewed and approved your PR, it can be merged in master.
    - You can also help out by reviewing the pull requests of other people
    - Please keep an eye on your pull request afterwards, the reviewers may have questions or comments about it, or ask you to tackle things in a different way, before they can approve it
    - Most discussions about the task or the pull request can happen through the github & waffle board comments.
    - If it's more complex you can bring it up in one of these meetings.
    - After your Pull Request has been merged, drag the issue to the done column on the waffle board. (this can also be automated by adding "this merge closes #{the id of the issue}" in the merge comment.

Documentation

- We'll need information on the wiki to help people learn angular 2 and universal development
    - Monika Mevenkamp will get started on that
    - Art Lowel (Atmire) will send her a list of links
- We'll need to properly document the development process on the wiki, perhaps starting from the "how to contribute" list above
- Our best practices and code standards also need to be added.
- We need a complete list of features we aim to achieve in DSpace 7 listed publicly, not just on the waffleboard
    - Perhaps we can ask DCAT to help with this
    - This is an opportunity to also link up features with the Use Cases that DCAT gathered. Some use cases may be easily implemented in Angular that were NOT as easily achieved in existing UIs
    - We can then more easily draw from this master list to create new tickets in Waffleboard for each work package

Second Hour: REST meeting