Updates on UI Selection Process

Update June 20, 2016: The selection of the Angular 2 UI platform was announced and the proof-of-concept Angular 2 UI was demoed at the Open Repositories conference in Dublin.

- Slides: http://www.slideshare.net/tdonohue/introducing-the-new-dspace-user-interface
- Video: https://www.youtube.com/watch?v=ibxQ60wPjXs
- Ongoing Video Updates: https://www.youtube.com/playlist?list=PLER-v9UmlFBut2SVIA2uUSuBFhSFnqIXh

Update May/June, 2016: We are now providing video updates on the Angular 2 UI prototype. Please visit our YouTube Playlist to catch up on all the updates. At the Open Repositories conference in Dublin, we will also be providing a live update and demonstration during the "You heard it here first" session on Weds, June 15 at 2:00pm.

Update April, 2016: The extended prototype on Angular 2 (see below) has begun at:

- https://github.com/DSpace-Labs/angular2-ui-prototype
- Details about the extended prototype can be found in its README and Wiki pages

Update March 24, 2016: After detailed discussion of all prototypes, public feedback, etc, the UI Working Group has written a detailed Summary of options / discussions:

- DSpace UI Working Group Summary: https://docs.google.com/document/d/1bvJrRWEOzLTCqlQYaJ_XSGZKHlXJmgHht/ZYGek4/edit

As detailed in this document, the UI architecture has been narrowed down to two options under consideration:

1. A Java-based, server-side solution based on Spring Boot. For a sample, see Prototype #1 below.
2. A Javascript-based, client-side solution using Spring Boot (for server-side components) and likely Angular 2 (for the client-side UI). Prototypes #7 and #4 featured client-side UI options, but neither used Angular 2.

At the DuraSpace Summit (March 16-17), these options were discussed with attendees from the DSpace Steering and Leadership groups. During that discussion, it was decided that we would move forward with an extended prototype of the client-side solution featuring Angular 2. As noted, none of our original prototypes used Angular 2 (though one did use Angular 1). Yet, Angular 2 has great interest in that it claims to solve the major SEO (search engine optimization) shortcomings of Angular 1. All original client-side prototypes (#7 and #4) had these SEO shortcomings, which made it difficult to judge their capabilities as a future DSpace UI.

The extended prototype phase will move into May. At that time, a final decision between the two platforms will be made.
DSpace User Interface Prototype Challenge

Help us discover the technology/platform for our new user interface (UI) for DSpace! Create a prototype UI on a platform of your choice (in any programming language), with basic DSpace-like capabilities as described below. The goal of a UI prototype is to exercise a new UI technology/platform to see whether it would meet the needs of our DSpace community.

- **Why?** In time for DSpace 7.0 (late 2016), the plan is to develop a new (single) UI for DSpace that will replace our aging JSPUI and XMLUI interfaces. (See RoadMap)
- **How?** The goal of the “UI Prototype Challenge” is to give everyone a chance to show off what you feel may make a great replacement UI framework. Have a cool idea for what would make a better DSpace UI? This is your chance to show it off in 80 hours or less! Register your idea on our sign-up page (see below) to find other collaborators.
- **What?** Prototypes are not meant to be full applications, but more of a proof-of-concept (something that proves a new DSpace UI could be built in your technology or platform of choice). Prototypes need not be Java (any/all languages welcome). Additional guidelines/scope is provided below.
- **When?** Prototypes / proof-of-concepts are due by **Friday, Nov 6 to Friday, Dec 4**. At that time, all will undergo a public review period. We will work with Committers, DCAT, Steering and Leadership Groups to select a “best of class” platform to build the new DSpace UI. Even incomplete proof-of-concepts are useful to submit, as we are judging the suitability of the “platform” and not the flashiness of the prototype itself.
- **What do I win?** While there are no official prizes (at this time), this challenge gives you an excellent opportunity to increase your reputation in our DSpace Community. It also provides you and your institution with an opportunity to directly influence the path of the DSpace UI going forward.
- **Other questions?** Ask them on the dspace-devel mailing list, and we’ll be glad to clarify any of the goals/guidelines of this challenge.

Good luck! The full guidelines / requirements are included below.

Guidelines

- **Prototypes are due by Friday, Dec 4.** Please sign up below to register your prototype (and possibly find collaborators).
- **Do not spend more than 80 hours total per prototype!**
- **Use the REST API or DSpace Java API (v5 or pre-6) when accessing DSpace content. Direct database or bitstream file system access is not allowed.**
- **Use Bootstrap or emulate the existing XMLUI Mirage2 theme.** Prototype UI design will not be a major factor in the decision, so keep it simple. Focus on functionality over fancy UI look and feel.
- **Make sure the prototype is “developer friendly”:** keep the code readable and understandable at this early point. Keep the code DRY (Don’t Repeat Yourself), but don’t overdoo it.
- **Make all code publicly available on GitHub, along with installation instructions.** A publicly accessible demo is also nice but not required.
- **Documentation may be provided as a GitHub README, or via a publicly accessible Wiki page (feel free to use the DSpace Wiki).**
- **Features that can’t be implemented in 80 hours or features not mentioned below may be mocked up or wireframed. But again, please don’t spend more than 80 hours.**
• NOTE: We will be evaluating the fit of the prototype’s platform and not the flashiness or even completeness of the prototype itself. For example, an incomplete prototype revealing a lot of potential for the platform may be preferable to a complete prototype that reveals platform-level limitations.

Prototype Scope

The following features should be implemented or mocked up in all prototypes. These basic features are meant to "exercise" some very basic DSpace UI needs within the selected UI platform/technologies.

1. **User Interface Layout:** The prototype UI should also fulfill these general layout requirements:
   a. The prototype should have a header and a footer.
   b. The prototype should have something to help an end user keep track of their location in the system (e.g., a breadcrumb trail).
   c. The prototype should have a menu or sidebar that contains context-sensitive options.
   d. The prototype should allow users to go to the main locations in the system using a url: the browser history, back, and previous buttons, should all work.
   e. The prototype should display and function well on screens of all sizes (responsive / mobile friendly, not native mobile).

2. **Simple Item View:** The prototype should have an equivalent of a simple item view, which displays key metadata about the item, along with links to downloadable files.

3. **Community / Collection View:** The prototype should have an equivalent to a Community/Collection view page, which displays key metadata about a community or collection.

4. **Browse Navigation:** The prototype should allow basic browse navigation of communities and collections and items.

5. **Authentication:** Users should be able to authenticate themselves via simple password authentication.

6. **Authorization:** A very basic example of authorization operations should be supported. For example, only allow editing of items to authenticated users in the Administrator group.

7. **Edit/Create Item:** The prototype should offer a basic capability to edit or create items via a simple web form. (Please no submission workflows as that is out of scope!) A single page form can serve both edit and create purposes. Not all metadata fields need be included, we are just looking for a simple example of a web form, including the ability to upload a file.

Customization Capabilities

A new DSpace UI should allow for the following common UI customization options. Based on your prototype, please either show an example of how these may be achieved in this UI platform or describe them via documentation.

1. Show or describe how an administrator would be able to easily adjust the site wide theme layout based on local design/needs. Specifically, show or describe how the following changes might be achieved:
   a. How would someone change the colors, fonts, sizes of the site? (e.g. css changes)
   b. How would someone modify the sitewide header/footer? (e.g. to change logo, title, etc)
   c. How would someone change the location of the breadcrumb trail (e.g. from header to footer)?
   d. How would someone add or remove links in the navigation menu (e.g. links to other university pages)?

Modularization Capabilities

A new DSpace UI should allow for / support common modularization needs. Based on your prototype, please describe (via documentation) how you feel this UI platform may (or may not) be able to achieve the modularization examples below:

1. How could this UI platform support optional modules/features?
   a. For example, Embargo is an existing optional feature within DSpace. While it is disabled by default, many sites choose to enable it.
   b. Enabling Embargo functionality requires additional metadata fields to be captured in the deposit form (e.g. embargo date, embargo reason).
   c. Does this UI framework support the idea of easily enabling (or disabling) features that require UI-level changes? In other words, in this framework, do you feel enabling Embargo functionality could be automated via a configuration (e.g. embargo.enabled=true)? Or would it require manual editing of the deposit form to add additional metadata fields?

2. How could this UI platform support new extensions/add-ons?
   a. Assume that someone has created a new Streaming feature for DSpace which provides a streaming capability for Video/Audio content. How might this UI platform support overriding/overlaying the default Item View to make that streaming feature available?

Prototype Documentation

Each prototype MUST be submitted with the following documentation. Some of this documentation includes brainstorming on how you feel this prototype or UI platform might be extended to support more advanced UI features. Please base your answers on what you know (or have learned) about this UI platform during the prototyping process.

1. Describe the design of the prototype (e.g. technologies/platforms used, including version of DSpace, etc.)?
2. How do you install the prototype on a new system? (Note: we will be testing the installation of prototypes in order to evaluate the installation of the platform itself)
3. How would you envision i18n (internationalization) support could be added to this UI prototype/platform in the future?
4. How would you envision theming capabilities could be added to this UI prototype/platform in the future? In other words, how might local or third-party themes be installed or managed? Think of a theme as a collection of styles, fonts, logo, and page overrides.
5. How would you envision supporting common DSpace authentication mechanisms (e.g. LDAP, Shibboleth) in this UI prototype/platform in the future?

Out of Scope

The following DSpace features should be considered out-of-scope, and need NOT be implemented in prototypes. No bonus will be given for implementing any of these features:

- Search capabilities (especially via Solr)
- Submission or Approval Workflows.
- Administrative UI
- Any other DSpace features not specifically mentioned in the “Prototype Scope” section above.

Prototype Sign-Up

<table>
<thead>
<tr>
<th>UI Platform / Technologies</th>
<th>Open Source License(s)</th>
<th>Technology Resources</th>
<th>Prototype Links / Notes</th>
<th>Prototyping Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>What UI platform /technologies are you prototyping? (e.g. Java Play!, Ruby on Rails, etc)</td>
<td>What Open Source license(s) is this technology/platform released under?</td>
<td>Add links to resources /documentation for any technologies that this prototype uses</td>
<td>Link to either a public wiki page describing your prototype, or to a public README in GitHub. Keep in mind, your codebase must also be public.</td>
<td>List your team members</td>
</tr>
<tr>
<td>Spring Boot + Thymeleaf</td>
<td>• Apache 2 (Spring Boot, Thymeleaf) • MIT (Bootstrap)</td>
<td>• Spring Boot (UI framework) • Thymeleaf template engine (builds the HTML) • Bootstrap (for CSS styling THEMING) • DSpace Java API (pre-6 version) • Spring Security (Authentication/Authorization)</td>
<td>Codebase: <a href="https://github.com/tdonohue/DSpace-Spring-Boot">https://github.com/tdonohue/DSpace-Spring-Boot</a> (see specifically the new ‘dspace-ui’ module) Documentation/Challenge Questions in the README: <a href="https://github.com/tdonohue/DSpace-Spring-Boot/blob/spring-boot-ui/README.md">https://github.com/tdonohue/DSpace-Spring-Boot/blob/spring-boot-ui/README.md</a> Prototype does not yet implement editing/deposit, so you need to have content added from elsewhere. (Sorry, I didn't have as much time as I hoped to have to work on this) Prototype Slides: <a href="http://www.slideshare.net/tdonohue/dspace-ui-prototype-challenge-spring-boot-thymeleaf">http://www.slideshare.net/tdonohue/dspace-ui-prototype-challenge-spring-boot-thymeleaf</a></td>
<td>Tim Donohue</td>
</tr>
<tr>
<td>Java API + Jersey + Twirl (in Scal)</td>
<td>• CDDL 1.1 (Jersey) • Apache 2.0 (Twirl)</td>
<td>• Java API • Jersey • Twirl</td>
<td>Codebase: <a href="https://github.com/Cineca/framework-lite">https://github.com/Cineca/framework-lite</a> Source code: <a href="https://github.com/Cineca/DSpace/tree/dspace-prototype-ui">https://github.com/Cineca/DSpace/tree/dspace-prototype-ui</a> Installation procedure: as a &quot;standard&quot; DSpace 6 JSPUI, details on the README of the project Documentation: Cineca UI Prototype - Documentation Presentation Slides: cineca-dspace-ui.pptx</td>
<td>Andrea Bolli (4Science) Cineca developers - Luigi Andrea Passarelli (4Science), Unknown User (r.suardi)</td>
</tr>
<tr>
<td>Spring MVC + Spring tags + Security + SiteMesh + custom extensions</td>
<td>• Apache 2 (Spring) • OpenSymphony Compatible with the Apache License (SiteMesh) • MIT (Bootstrap)</td>
<td>• Spring MVC &amp; Spring Tags • Spring Security • SiteMesh</td>
<td>Codebase: <a href="https://github.com/Cineca/DSpace/tree/dspace-prototype-ui">https://github.com/Cineca/DSpace/tree/dspace-prototype-ui</a> Installation procedure: as a &quot;standard&quot; DSpace 6 JSPUI, details on the README of the project Documentation: Cineca UI Prototype - Documentation Presentation Slides: cineca-dspace-ui.pptx</td>
<td>Jacob Brown (others welcome)</td>
</tr>
</tbody>
</table>
This prototype will leverage an existing Spring Boot and Angular code base that is being used in a variety of recent projects including the next version of the Vireo ETD management system.

**Demo:** http://library.tamu.edu/dspace-ui/

**Source:** https://github.com/TAMULib/DSpace-UI-prototype

**UPDATE - Official release of Core UI module:** https://github.com/TAMULib/Weaver-UI-Core

**UPDATE - Official release of Core Webservice dependency:** https://github.com/TAMULib/Weaver-Webservice-Core

**Slides:** DSpace-UI-Prototype-Presentation.pptx

---

### Prototype Presentation Schedule

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Team /Presenter</th>
<th>Prototype #</th>
<th>Recording of Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thurs., Jan 14, 10:00am EST</td>
<td>Art Lowel (Atmire)</td>
<td>#4 (REST + Ember.js)</td>
<td>(NOTE: We encountered technical difficulties during the live session, and the original recording had no audio. Therefore, the above video is a demo/screencast recorded by Art at a later time.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Presentation Slides: <a href="http://www.slideshare.net/bramluyten/dspace-ui-prototype-deember">http://www.slideshare.net/bramluyten/dspace-ui-prototype-deember</a></td>
</tr>
<tr>
<td>Mon., Jan 18, 10:00am EST</td>
<td>Peter Dietz</td>
<td>#2 (REST + Ruby on Rails), #3 (REST + Play), #9 (Rails + dspace-rest-client gem)</td>
<td></td>
</tr>
<tr>
<td>Thurs., Jan 21, 10:00am EST</td>
<td>Texas A&amp;M Libraries (James Silas Creel)</td>
<td>#7 (REST + Spring Boot + Angular.js)</td>
<td>Presentation Slides: DSpace-UI-Prototype-Presentation.pptx</td>
</tr>
<tr>
<td>Date</td>
<td>Time</td>
<td>Presenters</td>
<td>Topic</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mon., Jan 25</td>
<td>10:00am EST</td>
<td>Monika Mevenkamp, Ekaterina Pechekhonovarina</td>
<td>#8 (Rails + Bootstrap)</td>
</tr>
<tr>
<td>Thurs., Jan 28</td>
<td>10:00am EST</td>
<td>Unknown User (r.suardi), Luigi Andrea Pascarelli (4Science)</td>
<td>#6 (Spring MVC + Spring tags + Spring security + sitemesh + custom extensions)</td>
</tr>
<tr>
<td>Monday, Feb 1</td>
<td>10:00am EST</td>
<td>Tim Donohue</td>
<td>#1 (Spring Boot + Thymeleaf)</td>
</tr>
<tr>
<td>Thursday, Feb 4</td>
<td>10:00am EST</td>
<td>Jacob Brown</td>
<td>#5 (Java API + Jersey + Twirl (in Scala))</td>
</tr>
</tbody>
</table>