DSpace UI Prototype Challenge

- DSpace User Interface Prototype Challenge
-
    - Guidelines
    - Prototype Scope
    - Customization Capabilities
    - Modularization Capabilities
    - Prototype Documentation
    - Out of Scope
- Prototype Sign-Up
    - Prototype Evaluation Process
    - Prototype Presentation Schedule

Next phase of this work (creating the DSpace 7 UI) has moved over to DSpace 7 Working Group (2016-2023)
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=lbxQ60wjPXs
- Ongoing Video Updates: https://www.youtube.com/playlist?list=PLER-v9UmiFBut2SVfA2uUSuBFhSFnqlXh

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/1bvJrRWEO2ZLTCLqNiaJ_KXGZKHNfXDjnqHWt_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.
Evaluation Timeline / RoadMap

How did we get here? Here's the timelines of where we've been and where we are going

- Dec 4: DSpace UI Prototype Challenge ends and review period begins
- Jan - Feb: Demonstrations/discussions of all submitted prototypes. Video demos and discussions available below
- *Week of Feb 8-12:* Collect oral and written feedback on prototypes, focusing on "Advantages, Concerns & Questions" (see call for feedback above and our public evaluation form). A public call for feedback will be sent to all mailing lists once all prototype videos are available/posted.
- *Week of Feb 15-19:* Resolve outstanding questions. Select a shortlist of 2-3 architectures for consideration.
- *Week of Feb 22-26:* Select no more than 2 finalist architectures, which might include combinations of technologies from the various prototypes
- *Week of Feb 29-March 4:* ~~Select an architecture for recommendation;~~ validate it against our criteria (see evaluation form). See UI Working Group Summary.
- *March 16-17:* Presentation of recommended architecture to DSpace Steering and Leadership groups at the DuraSpace Summit Meeting for their feedback / approval.
- March / April : Extended prototype phase of an Angular 2 prototype as recommended at DuraSpace Summit. A team comprised of several institutions (Texas A&M, @mire, Cineca, DuraSpace) will collaborate on this prototype, and determine its plausibility (as a future UI) and whether it meets our SEO needs.
- May / June: Continued prototype (Phase II) of Angular 2 prototype in preparation for a demonstration at Open Repositories 2016.

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~~ **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 overdo 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 adjust the navigation bar to appear on left or right?
    d. How would someone change the location of the breadcrumb trail (e.g. from header to footer)?

e. How would someone display additional metadata fields on the Item view page? For example, displaying “dc.publisher” on the default Item view site wide?
f. How would someone add new links or an entire section to 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 brainstorms 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

UI Platform / Technologies	**Open Source License(s)**	**Technology Resources**	**Prototype Links / Notes**	**Prototyping Team**
What UI platform /technologies are you prototyping? (e.g. Java Play!, Ruby on Rails, etc)	*What Open Source license(s) is this technology/platform released under?*	*Add links to resources /documentation for any technologies that this prototype uses*	*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.*	*List your team members*
Spring Boot + Thymeleaf	• Apache 2 (Spring Boot, Thymeleaf) • MIT (Bootstrap)	• 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)	• **Codebase:** https://github.com/tdonohue/DSpace-Spring-Boot (see specifically the new 'dspace-ui' module) • Documentation/Challenge Questions in the README: https://github.com/tdonohue/DSpace-Spring-Boot/blob/spring-boot-ui/README.md • 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:** http://www.slideshare.net/tdonohue/dspace-ui-prototype-challenge-spring-boot-thymeleaf	Tim Donohue

REST + Ruby on Rails	MIT License (Ruby on Rails)	• DSpace REST • http://rubyonrails.org/	• Codebase: https://github.com/peterdietz/dspace-rest-rails • Demo site: http://dspace-rails.herokuapp.com/ • Video of discussion session: https://www.youtube.com/watch?v=N35yhaMCvL4	Peter Dietz
REST + Play!	Apache 2 (Play Framwork)	• DSpace REST • https://www.playframework.com/	• Codebase: https://github.com/peterdietz/dspace-rest-play • Demo site: http://dspace-rest-client-play.herokuapp.com/ • Video of discussion session: https://www.youtube.com/watch?v=N35yhaMCvL4	Peter Dietz
REST + Ember.js	MIT License (Ember.js)	• DSpace REST • http://emberjs.com/ • http://ember-cli.com/	**Demo:** http://dsember.atmire.com/ **Source code:** https://github.com/atmire/dsember https://github.com/atmire/dsember-core **Answers to challenge questions:** https://github.com/atmire/dsember/wiki	Art Lowel (Atmire)
~~Java API + Jersey + Twirl (in Scala)~~ *Prototype was withdrawn by its author because of lack of time to work on it.*	• CDDL 1.1 (Jersey) • Apache 2.0 (Twirl)	• Java API • Jersey • Twirl	https://github.com/kardeiz/disco	Jacob Brown (others welcome)
Spring MVC + Spring tags + Spring security + sitemesh + custom extensions	• Apache 2 (Spring) • OpenSymphony Compatible with the Apache License (Sitemesh) • MIT (Bootstrap)	• Spring MVC & Spring Tags • Spring security • SiteMesh	This prototype will rely on custom extensions build by Cineca on top of the selected technologies (see https://github.com/Cineca/framework-lite). These extensions aim to speed up the development time providing lot of reusing and configurable components. Source code will be provided as part of the prototype **Source code:** https://github.com/Cineca/DSpace/tree/dspace-prototype-ui **Installation procedure:** as a "standard" DSpace 6 JSPUI, details on the README of the project **Documentation:** Cineca UI Prototype - Documentation **Presentation Slides:** cineca-dspace-ui.pptx	Andrea Bollini (4Science) Cineca developers - Luigi Andrea Pascarelli (4Science), Unknown User (r.suardi)
REST + Spring Boot + Angular.js	• Apache 2 (Spring Boot) • MIT (Angular.js, Bootstrap, sass)	• Spring Boot (webapp, models, controllers) • websockets • Bootstrap and Sass (for CSS styling/theming) • Angular.js (views)	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**	Texas A&M University Libraries (James Silas Creel)
Rails + Bootstrap	MIT	Rails http://rubyonrails.org/ Bootstrap http://getbootstrap.com/	Levering gems such as devise for authentication. **MCM Source Rails App:** github.com/akinom/dspace-rails **MCM Demo**: At Heroku **EKate Rest Code**: github.com/ekate/dspace-rest-rails which leverages the gem active-rest-client **Ekate Blacklight Code:** https://github.com/ekate/blacklight-dspace **Presentation Slides**: @google	Monika Mevekamp monikam@princeton.edu Ekaterina Pechekhonova ekate@nyu.edu
Rails + dspace-rest-client gem		DSpace REST Rails dspace-rest-client gem (https://gitlab.c3sl.ufpr.br/c3sl/dspace-rest-client)	Using the dspace-rest-client gem within rails to very simply and cleanly access dspace objects. In this project I don't deal with parsing json/xml, its all a simple gem library. i.e. @items = @client.items.find(id: => params[:id]) • Code: https://github.com/peterdietz/drail • Demo: http://drail.herokuapp.com/ • Video of discussion session: https://www.youtube.com/watch?v=N35yhaMCvL4	Peter Dietz

Prototype Evaluation Process

The DSpace UI Prototype Working Group (2015-16) will be working with prototype developers to schedule a recorded Q&A discussion for each prototype (starting in January 2016). These Q&A session recordings will be made publicly available on this wiki page.

The Working Group has also create a public Prototype Evaluation Form, where all individual or group evaluations will be recorded. Individuals are also welcome to provide your feedback on specific prototypes via this Evaluation Form. The Evaluation Form is a Google Spreadsheet, where each prototype has its own tab for feedback/evaluation:

- **Prototype Evaluation Form:** https://docs.google.com/spreadsheets/d/1XWxEERh0UXCOo7LhocMMaqbY2u4b6oI11oWOjSp_aSQ/edit

Prototype Presentation Schedule

Please provide your feedback / evaluation!

Anyone is welcome to enter your personal (or team) evaluation for any of the Prototypes in our public Prototype Evaluation Form. Each prototype has a separate evaluation tab in this form. We ask that you add your comments (advantages / concerns / questions) to the "Public Comments" section for each prototype.

https://docs.google.com/spreadsheets/d/1XWxEERh0UXCOo7LhocMMaqbY2u4b6oI11oWOjSp_aSQ/edit

Date/Time	Team /Presenter	Prototype #	Recording of Session
Thurs., Jan 14, 10:00am EST	Art Lowel (Atmire)	#4 (*REST + Ember.js*)	(*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.*) **Presentation Slides:** http://www.slideshare.net/bramluyten/dspace-ui-prototype-dsember
Mon., Jan 18, 10:00am EST	Peter Dietz	#2 (*REST + Ruby on Rails*), #3 (*REST + Play!*), #9 (*Rails + dspace-rest-client gem*)	
Thurs., Jan 21, 10:00am EST	Texas A&M Libraries (James Silas Creel)	#7 (*REST + Spring Boot + Angular.js*)	**Presentation Slides:** DSpace-UI-Prototype-Presentation.pptx
Mon., Jan 25, 10:00am EST	Monika Mevenkamp, Ekaterina Pechekhonovarina	#8 (*Rails + Bootstrap*)	**Presentation Slides**: @google
Thurs., Jan 28, 10:00am EST	Unknown User (r.suardi), Luigi Andrea Pascarelli (4Science)	#6 (*Spring MVC + Spring tags + Spring security + sitemesh + custom extensions*)	**Presentation Slides:** cineca-dspace-ui.pptx
Monday, Feb 1, 10:00am EST	Tim Donohue	#1 (*Spring Boot + Thymeleaf*)	**Presentation Slides:** http://www.slideshare.net/tdonohue/dspace-ui-prototype-challenge-spring-boot-thymeleaf
~~Thursday, Feb 4, 10:00am EST~~	~~Jacob Brown~~	~~#5 (*Java API + Jersey + Twirl (in Scala)*)~~	*Prototype was withdrawn by its author because of lack of time to work on it.*