UI roadmap and prototype

Tim Hostetler and the ADS Team

ADS Users Group Meeting, 15-16 Nov. 2021

Current UI Status

- ADS UI (Bumblebee)
    - Single page application
        - JavaScript required
    - Older tech stack
- ADS Core
    - For slow devices
    - Limited workflow
    - Server-rendered (static)
    - No JavaScript

Goals

- Modernize tech stack
    - NextJS
    - ReactJS
    - 100% Typescript
- Improve developer experience
    - Easier maintenance
    - Easier to add/change features (modular)
- Remove the need for two separate systems to maintain

Success Criteria

- Faster loading times
- Usable in slow devices and/or slow internet connection
- Server-side rendered content for crawlers without JavaScript (e.g., GoogleBot)
- Accessible from scratch
- Responsive from scratch (computer, tablet, phone…)
- Consider ADS expansion needs from scratch (e.g., themes, moving away from bibcodes)
- Do not break completely the current user experience
- Improve the user experience based on user feedback and user studies

Implementation

- Hybrid server-side/client rendering
  - Initial request is server-rendered (per route).
  - Subsequent requests are client-side triggered
- Initial experimental design
  - Plan to have prototype(s) for a round of user testing in the future

Initial Milestones

- Reach functional parity with ADS Core
  - Basic search ability
  - Abstract pages with all subpages
  - Export single records
  - Basic metrics
  - Non-js limited functionality

Demo

[ADS Core](#)

[Nectar](#)

Work to do

- Continue to add features to match ADS' functionality
  - Facets/filters
  - Authenticated features
    - Libraries
    - User settings