CHALIKI

a desktop Scripture browser

powered by Proskomma

Design

- Electron boilerplate
- React with
  - Functional components
  - useEffect for async queries
  - prop threading
  - Material UI

Design

- Load all content at startup
- Shared navigation state
- One tab per experiment sharing common data

Design – next steps

- Agree on Proskomma React best practice
- Beat Electron into submission
- Refactor to allow plugin tabs

DocSets

- 14 translations (817 books)
- Loading from Proskomma JSON
  - 483 Mb unpacked for 14 translations
  - 338 Mb unpacked for 4 uW translations
  - 45 Mb for 4 uW translations in zip file

DocSets – next steps

- Load via mutations – async
- Allow manual load/unload
- Support loading from remote data sources

Browse

- Intelligent navigation
- Verse view
- Chapter view, with paragraph styles
- Block view, with paragraphs
- Clickable verse numbers

Browse – next steps

- Render all formatting (\nd, \add…)
- Optionally render grafts
- Explore “click on word’ functionality

Search

- Multiple terms with and/or logic
- Each term is a regex
- Search time close to linear regardless of number of terms specified or found
- Pagination of results (reduces React overhead, still fetches all results)

Search – next steps

- Search on exact term (already in graph)
- Fetch one book at a time
- Search on lemma, source word…
- Would be faster in Go...

Verse Mapping

- Finds equivalent verse content regardless of different BCV in different translations
- Handles most scenarios – one known, unhandled edge case
- Requires versification for all included translations

Verse Mapping – next steps

- Fix edge cases
- Alignment across verse mapping
  - Currently no translations with both versification information and uW alignment data
- Expose local BCV for mapped translations (needs to get past GraphQL types)