Scripture as Graph Proof of Concept

By Mark Howe
July 21st 2020

Two Loosely-Coupled Issues

- Peer to Peer Syncing
  - Assumes GunDB
- Token/Graph Representation of Scripture
  - Could use something like GunDB

GunDB for Tokenized Documents

- Extremely Stateful!
- Limited Storage on Client (5Mb?)
- Integrity issues with lots of tokens:
    - Timeouts due to blocking main node thread
    - Requires rewriting “Every Loop in GunDB”
    - No timescale for doing this
    - May not address scalability issues
    - Limited documentation on, eg, scalability issues

Building a Graph

- ProtoTokens
- Smart Tokens
- Lookups

ProtoTokens

```
        [
            "cv",
            "([\\r\\n]*\\\\[cv][ \\t]\\d+[ \\t\\r\\n]*)",
            "[\\r\\n]*\\\\([cv])[ \\t](\\d+)[ \\t\\r\\n]*"
        ],
        [
            "attribute",
            "(\\|?[A-Za-z0-9\\-]+=\"[^\"]+\"[ \\t]?)",
            "\\|?([A-Za-z0-9\\-]+)=\"([^\"]+)\"[ \\t]?"
        ],
...
```

Tokens

- Stored as objects, accessed by tokenID
- Linked lists for
  - Body
  - Header
  - Heading
  - REM
  - Note
- (Ignore attributes for now)

Tokens

- Reconstruct text backwards
- Lookups (start/end):
    - Paragraphs
    - chapter/verse
    - Span and word-level markup
    - ...

Smart Tokens

- Words
- Paras
- (Strongs, span markup...)

The Data Set

- English
  - ULT
  - WEB
- French
  - LSG

(Error in ULT NEH)

CLI Utility

```
node sag.js sag_usfm/eng/ult/psa.usfm verse 119 105
Init in 3228 msec

TEXT FOR ONE VERSE
Your word is a lamp to my feet and a light for my path.

Query in 2 msec
60 Mb used after query
```

Node Server

- 2 Endpoints:
  - Available content (nested objects, grep’d from directory)
  - Doc as USFM

- Running on port 4000

React Client

- Bootstrap

- Running on port 3000

Options for Future Work

- Better Parsing
- Better Model
- Better System Architecture

Better Parsing

- Optimize
  - Leaner Regexes
  - Real Pointers
- Handle Attributes
- More Careful Tag Handling
  - Milestones?
- Error Handling/Validation

Better Model

- Smarter Tokens
- Translation-level Representation
- Classes

Better Architecture

- Serialized Representation of Tokens
- Tokenize on Server
- Parse on Demand?
    - Tree Bloat
- Use Document DB?