Collect & Distribute: Utilizing the APIs of GOKb

Moritz Horn
WOLFcon 2022
Components in GOKb
GOKb APIs

- OAI-PMH (harvesting)
- Opensearch
- KBART export
- REST
OAI-PMH

- Metadata formats „dc“ & „gokb“
  - custom „gokb“ format retrieves full info with (directly) linked items
  - Harvesting with temporal filters (‘from‘ & ‚until‘) & pagination via ‚resumptionToken‘

- Separate endpoints for each component
  - packages (identifiers, context, provider, platform, TIPPs)
  - titles (identifiers, name variants, TIPPs)
  - providers (identifiers, name variants, platforms)
  - platforms (name variants, url, provider)
  - tipps (url, coverage, access, package, platform, title)
OAI-PMH - Packages

- Contains all info about the package itself ([link](#))
- Links to provider & platform
- Complete info about contained TIPPs
  - Coverage
  - Access
  - Basic title info
- Possibly large XML-Objects, has to be cached
  - Delayed visibility after import/update
  - Much unchanged info gets reprocessed
  - Each manual TIPP update triggers caching overhead
OAI-PMH - TIPPs

- Single package content items ([example](#))
- Links to package, platform, title instance
- Coverage, access, identifiers
- Additional filtering by package
OAI-PMH – Title Instances

- Full reference title info (example)
- Links to Publisher(s) & TIPPs
- Identifiers, name variants, publication dates & history
- Not affected by TIPP changes
OAI-PMH - Providers/Orgs

- Full provider info (example)
- Provided platforms & packages
- Identifiers & variant names
Opensearch FTI Search

- Different endpoints for several use cases
  - Targeted search
  - Scroll API for retrieval of big result sets
- Much more streamlined JSON result records
- Indexing delay to be considered
- Single endpoint targeting separate indices via param (‘componentType‘)
Targeted Search

- Filtering by most important properties (example)
  - Linked items (i.e. TIPPs of package, publisher etc)
  - Temporal & Status filters
  - Search for (several) identifiers
  - Strict name matching & ngram search for autocomplete
Scroll API

- Temporal filtering (‘changedSince’ & ’changedBefore’) 
- Status filtering 
- Maximum of records per response 
  - bigger sets return ’scrollId‘ for result set pagination
KBART export

- Endpoint for retrieving packages as they came in
- Caching necessary same as for OAI-PMH XML
- Two different flavors
  - *processed*: Values are selected preferably from referenced & enriched title instances
  - *raw*: Values are preferably taken from incoming KBART (TIPP)
REST

- Endpoint used for frontend communication
- Most data retrieval calls without authentification
- Create & update via bearer token
- Dedicated endpoint for ingesting KBART, watching import jobs
More Info

- Home page & documentation: https://gokb.org
- User interface: https://gokb.org/gokb-ui
- Refs: https://github.com/openlibraryenvironment/gokb/wiki/API#gokb-apis