Package: openalexSnapshot (via r-universe)

June 2, 2026

Title OpenAlex Bulk Snapshot Conversion, Indexing, and Record Extraction

Version 0.0.2

Author Rainer M Krug

Maintainer Rainer M Krug <Rainer@krugs.de>

Description Provides tools for working with the OpenAlex bulk snapshot: converting .json.gz files to Parquet format, building ID-lookup indexes over the resulting corpus, and extracting records by OpenAlex ID. Large-scale operations delegate to a compiled Rust back-end (openalex-core via extendr); a pure-R/DuckDB fallback is included for environments without a Rust toolchain.

License GPL (>= 2)

URL https://github.com/openalexPro/openalexSnapshot, https://openalexpro.github.io/openalexSnapshot/, https://doi.org/10.5281/zenodo.20448992

BugReports https://github.com/openalexPro/openalexSnapshot/issues

Additional_repositories https://openalexpro.r-universe.dev

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0

Encoding UTF-8

Config/rextendr/version 0.3.1

Depends R (>= 4.1.0)

Imports arrow, dplyr

Suggests testthat (>= 3.0.0), pak

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake libssl-dev libclang-dev

Repository https://openalexpro.r-universe.dev

Date/Publication 2026-06-02 15:40:06 UTC

RemoteUrl https://github.com/openalexPro/openalexSnapshot

RemoteRef main

RemoteSha e6d7e5bfe1e781b93112005a585d54d439dcf943

Contents

`build_corpus_index` *Build a Parquet ID-lookup index*

Description

Builds a `<dataset>_id_idx.parquet` index from the Parquet corpus produced by `snapshot_to_parquet()`, enabling fast record retrieval by OpenAlex ID using `lookup_by_id()`.

Usage

```
build_corpus_index(
  root_dir = NULL,
  data_sets = NULL,
  workers = NULL,
  memory_limit = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  corpus_dir = NULL
)
```

Arguments

`root_dir`	Root directory containing a `parquet/` subdirectory produced by `snapshot_to_parquet()`. If provided, the index for each dataset in `data_sets` is created at `<root_dir>/parquet/<dataset>_id_i`
`data_sets`	Character vector of dataset names to index (e.g. `c("works", "authors")`). `NULL` indexes all datasets found under `<root_dir>/parquet/`. Ignored when `corpus_dir` is provided.
`workers`	Number of parallel workers for Stage 1 indexing. Default is `NULL` (sequential).
`memory_limit`	DuckDB memory limit (e.g., `"20GB"`). Default is `NULL`.

`overwrite`	If `TRUE`, rebuilds existing indexes. Default is `FALSE` (skip if the index already exists).
`verbose`	Print progress messages. Default is `TRUE`.
`corpus_dir`	Explicit path to a single dataset Parquet directory (e.g. `"/Volumes/openalex/parquet/works"`). The index is written as a sibling file: `<parent>/<basename>_id_idx.parquet`. When this is provided, `root_dir` and `data_sets` are ignored.

Details

The function uses a two-stage approach:

1. Index each Parquet file individually (bounded memory, parallel, with resume support).
2. Combine the per-file shard indexes into a single Parquet index.

Paths can be supplied as a single `root_dir` (which iterates over all requested `data_sets`) or as an explicit `corpus_dir` pointing to a single dataset directory.

The index contains columns:

id The OpenAlex ID

id_block Block number computed as `floor(numeric_id / 10000)`

parquet_file Relative path to the Parquet file in the corpus

file_row_number Row number within the file (0-indexed)

Value

When `corpus_dir` is provided, invisibly returns the path to the created index file. When `root_dir` is used, invisibly returns `root_dir`.

See Also

`snapshot_to_parquet()` for creating the Parquet corpus, `lookup_by_id()` for ID-based record retrieval.

Examples

```
## Not run:
build_corpus_index(root_dir = "/Volumes/openalex")

build_corpus_index(
  root_dir  = "/Volumes/openalex",
  data_sets = "works",
  workers   = 4
)

# Single explicit directory:
build_corpus_index(
  corpus_dir   = "/Volumes/openalex/parquet/works",
  memory_limit = "20GB"
)
```

```
## End(Not run)
```

---

lookup_by_id *Look up records by OpenAlex ID*

Description

Uses a pre-built index (created by build_corpus_index()) to locate records efficiently and extract them from the Parquet corpus.

Usage

```
lookup_by_id(
  root_dir = NULL,
  ids,
  project_dir = NULL,
  data_sets = NULL,
  workers = NULL,
  progress = TRUE,
  verbose = TRUE,
  index_file = NULL,
  selected = NULL,
  output = NULL
)
```

Arguments

root_dir	Root directory containing parquet/ and the dataset indexes produced by build_corpus_index(). Index files are expected at <root_dir>/parquet/<dataset>_id_idx.parquet.
ids	Character vector of OpenAlex IDs to retrieve. Can be long form (e.g. "https://openalex.org/W274180 or short form (e.g. "W2741809807").
project_dir	Project output directory. Extracted Parquet files are written to <project_dir>/snapshot_extract_<dat Only used when root_dir is provided.
data_sets	Character vector of dataset names to search (e.g. c("works", "authors")). NULL searches all indexed datasets under <root_dir>/parquet/. Ignored when index_file is provided.
workers	Number of parallel workers for reading corpus files. Default is NULL (sequential).
progress	Ignored (kept for backward compatibility).
verbose	Print progress messages. Default is TRUE.
index_file	Explicit path to an index Parquet file created by build_corpus_index(). When provided, root_dir, data_sets, and project_dir are ignored.

selected	Column selection passed to arrow::open_dataset(). Default is NULL (all columns).
output	Path to an output directory for writing results as Parquet files when using index_file mode. If NULL (default), results are returned as a data frame. Ignored when root_dir is used (use project_dir instead).

Details

Paths can be supplied as a root_dir + data_sets pair (which automatically locates the correct index files and writes output into project_dir) or as an explicit index_file for direct use.

Value

- index_file mode, output not NULL: invisibly returns output.
- index_file mode, output is NULL: returns a data frame of matching records.
- root_dir mode: invisibly returns project_dir.

See Also

build_corpus_index() for building the required index, snapshot_to_parquet() for creating the Parquet corpus.

Examples

```
## Not run:
# root_dir mode (searches multiple datasets)
lookup_by_id(
  root_dir    = "/Volumes/openalex",
  ids         = c("W2741809807", "W1234567890"),
  project_dir = "my_project",
  data_sets   = "works"
)

# index_file mode (direct access, returns data frame)
records <- lookup_by_id(
  index_file = "works_id_index.parquet",
  ids        = c("W2741809807", "W1234567890")
)

# index_file mode (write to parquet)
lookup_by_id(
  index_file = "works_id_index.parquet",
  ids        = large_id_vector,
  output     = "filtered_works",
  workers    = 3
)

## End(Not run)
```

oa_build_corpus_index *Build a two-stage ID-lookup index for a single Parquet corpus directory.*

Description

Stage 1: per-file shard indexes (parallel via rayon). Stage 2: combine shards into <corpus_name>_id_idx.parquet.

Usage

```
oa_build_corpus_index(corpus_dir, workers, memory_limit, overwrite, verbose)
```

Arguments

corpus_dir	Path to a single dataset Parquet directory.
workers	Number of parallel workers for Stage 1.
memory_limit	DuckDB memory limit ("" = no limit).
overwrite	If TRUE, rebuild an existing index.
verbose	Print progress to stderr.

Details

Returns the path to the created index file as a character scalar.

Value

Character scalar: path to the index file.

oa_lookup_by_id *Look up records by OpenAlex ID using a pre-built index.*

Description

Reads the index file, filters to the requested IDs, and extracts matching rows into the output directory (which must not already exist).

Usage

```
oa_lookup_by_id(index_file, ids, output, workers, verbose)
```

Arguments

index_file	Path to the index Parquet file (created by oa_build_corpus_index()).
ids	Character vector of OpenAlex IDs (long or short form).
output	Output directory path. Must not already exist.
workers	Number of parallel workers for file extraction.
verbose	Print progress to stderr.

Value

Invisibly returns NULL.

---

oa_snapshot_to_parquet

Convert an OpenAlex snapshot to Parquet format.

Description

Full pipeline: schema inference (per-dataset, cached in <parquet_dir>/<dataset>/.schema_cache/unified_schema.csv
plus parallel per-file COPY via rayon.

Usage

```
oa_snapshot_to_parquet(
  snapshot_dir,
  parquet_dir,
  data_sets,
  workers,
  sample_size,
  memory_limit,
  temp_dir,
  verbose
)
```

Arguments

snapshot_dir	Path to the snapshot root (contains a data/ subdir).
parquet_dir	Output directory for Parquet files.
data_sets	Character vector of dataset names, or character(0) for all datasets found under snapshot_dir/data/ (excluding merged_ids).
workers	Number of parallel workers (1 = sequential).
sample_size	Files to sample for schema inference (0 = all).
memory_limit	DuckDB memory limit, e.g. "8GB" ("" = no limit).
temp_dir	DuckDB temp directory ("" = system default).
verbose	Print progress to stderr.

Value

Invisibly returns NULL.

---

snapshot_to_parquet *Convert OpenAlex snapshot to Parquet format*

Description

Converts OpenAlex snapshot .json.gz files to Parquet using schema inference and parallel conversion. Paths can be supplied as a single root_dir (which derives snapshot_dir and parquet_dir automatically) or as explicit snapshot_dir and parquet_dir arguments.

Usage

```
snapshot_to_parquet(
  root_dir = NULL,
  data_sets = NULL,
  workers = NULL,
  sample_size = 20,
  memory_limit = NULL,
  temp_directory = NULL,
  progress = TRUE,
  verbose = TRUE,
  snapshot_dir = NULL,
  parquet_dir = NULL
)
```

Arguments

root_dir	Root directory. If provided, snapshot_dir defaults to <root_dir>/openalex-snapshot and parquet_dir defaults to <root_dir>/parquet.
data_sets	Character vector of dataset names to convert (e.g. c("works", "authors")). NULL converts all datasets found under <snapshot_dir>/data/.
workers	Number of parallel workers for file conversion. Default is NULL (sequential).
sample_size	Number of .gz files to sample for unified schema inference. Higher values give more accurate schemas but take longer. Default is 20. Use NULL or 0 to use all files.
memory_limit	DuckDB memory limit per worker (e.g., "8GB"). Default is NULL (DuckDB default).
temp_directory	Location of the temporary directory for DuckDB. Default is NULL (system default).
progress	Ignored (kept for backward compatibility).
verbose	Print per-dataset progress messages. Default is TRUE.

snapshot_dir	Explicit path to the snapshot data directory (the one containing a data/ subfolder). Required when root_dir is not provided.
parquet_dir	Explicit path to the Parquet output directory. Required when root_dir is not provided.

Value

Invisibly returns NULL.

See Also

build_corpus_index() for indexing the resulting Parquet files, lookup_by_id() for ID-based record retrieval.

Examples

```
## Not run:
snapshot_to_parquet(root_dir = "/Volumes/openalex")

snapshot_to_parquet(
  root_dir     = "/Volumes/openalex",
  data_sets    = c("authors", "works"),
  workers      = 4,
  memory_limit = "8GB"
)

# Explicit paths (no root_dir):
snapshot_to_parquet(
  snapshot_dir = "/data/openalex-snapshot",
  parquet_dir  = "/data/parquet",
  data_sets    = "authors"
)

## End(Not run)
```

Index